

Multimedia Applications of the Wavelet Transform

Inauguraldissertation zur Erlangung
des akademischen Grades eines
Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.–Math. oec. Claudia Kerstin Schremmer
aus Detmold

Mannheim, 2001

Dekan: Professor Dr. Herbert Popp, Universität Mannheim
Referent: Professor Dr. Wolfgang Effelsberg, Universität Mannheim
Korreferent: Professor Dr. Gabriele Steidl, Universität Mannheim

Tag der mündlichen Prüfung: 08. Februar 2002

*If we knew what we were doing,
it would not be called research, would it?*

— Albert Einstein

Abstract

This dissertation investigates novel applications of the wavelet transform in the analysis and compression of audio, still images, and video. In a second focal point, we evaluate the didactic potential of multimedia-enhanced teaching material for higher education.

Most recently, some theoretical surveys have been published on the potential for a wavelet-based restoration of noisy audio signals. Based on these, we have developed a wavelet-based denoising program for audio signals that allows flexible parameter settings. It is suited for the demonstration of the potential of wavelet-based denoising algorithms as well as for use in teaching.

The multiscale property of the wavelet transform can successfully be exploited for the detection of *semantic* structures in still images. For example, a comparison of the coefficients in the transformed domain allows the analysis and extraction of a predominant structure. This idea forms the basis of our semiautomatic edge detection algorithm that was developed during the present work. A number of empirical evaluations of potential parameter settings for the convolution-based wavelet transform and the resulting recommendations follow.

In the context of the teleteaching project *Virtuelle Hochschule Oberrhein*, i.e., Virtual University of the Upper Rhine Valley (VIROR), which aims to establish a semi-virtual university, many lectures and seminars were transmitted between remote locations. We thus encountered the problem of scalability of a video stream for different access bandwidths in the Internet. A substantial contribution of this dissertation is the introduction of the wavelet transform into hierarchical video coding and the recommendation of parameter settings based on empirical surveys. Furthermore, a prototype implementation of a hierarchical client-server video program proves the principal feasibility of a wavelet-based, nearly arbitrarily scalable application.

Mathematical transformations of digital signals constitute a commonly underestimated problem for students in their first semesters of study. Motivated by the VIROR project, we spent a considerable amount of time and effort on the exploration of approaches to enhance mathematical topics with multimedia; both the technical design and the didactic integration into the curriculum are discussed. In a large field trial on *traditional teaching versus multimedia-enhanced teaching*, in which the students were assigned to different learning settings, not only the motivation, but the objective knowledge gained by the students was measured. This allows us to objectively rate positive the *efficiency* of the teaching modules developed in the scope of this dissertation.

Kurzfassung

Die vorliegende Dissertation untersucht neue Einsatzmöglichkeiten der Wavelet-Transformation für die Analyse und Kompression der multimedialen Anwendungen Audio, Standbild und Video. In einem weiteren Schwerpunkt evaluieren wir das didaktische Potential multimedial angereicherter Lehrmaterials für die universitäre Lehre.

In jüngster Zeit sind einige theoretische Arbeiten über Wavelet-basierte Restaurationsverfahren von verrauschten Audiosignalen veröffentlicht worden. Hierauf aufbauend haben wir ein Wavelet-basiertes Entrauschungsprogramm für Audiosignale entwickelt. Es erlaubt eine sehr flexible Auswahl von Parametern, und eignet sich daher sowohl zur Demonstration der Mächtigkeit Wavelet-basierter Entrauschungsansätze, als auch zum Einsatz in der Lehre.

Die Multiskaleneigenschaft der Wavelet-Transformation kann bei der Standbildanalyse erfolgreich genutzt werden, um *semantische* Strukturen eines Bildes zu erkennen. So erlaubt ein Vergleich der Koeffizienten im transformierten Raum die Analyse und Extraktion einer vorherrschenden Struktur. Diese Idee liegt unserem im Zuge der vorliegenden Arbeit entstandenen halbautomatischen Kantensegmentierungsalgorithmus zugrunde. Eine Reihe empirischer Evaluationen über mögliche Parametereinstellungen der Faltungs-basierten Wavelet-Transformation mit daraus resultierenden Empfehlungen schließen sich an.

Im Zusammenhang mit dem Teleteaching-Projekt *Virtuelle Hochschule Oberrhein* (VIROR), das den Aufbau einer semi-virtuellen Universität verfolgt, werden viele Vorlesungen und Seminare zwischen entfernten Orten übertragen. Dabei stießen wir auf das Problem der Skalierbarkeit von Videoströmen für unterschiedliche Zugangsbandbreiten im Internet. Ein wichtiger Beitrag dieser Dissertation ist, die Möglichkeiten der Wavelet-Transformation für die hierarchische Videokodierung aufzuzeigen und durch empirische Studien belegte Parameterempfehlungen auszusprechen. Eine prototypische Implementierung einer hierarchischen Client-Server Videoanwendung beweist zudem die prinzipielle Realisierbarkeit einer Wavelet-basierten, fast beliebig skalierbaren Anwendung.

Mathematische Transformationen digitaler Signale stellen für Studierende der Anfangssemester eine häufig unterschätzte Schwierigkeit dar. Angeregt durch das VIROR Projekt setzen wir uns in einem weiteren Teil dieser Dissertation mit den Möglichkeiten einer multimedialen Aufbereitung mathematischer Sachverhalte auseinander; sowohl die technische Gestaltung als auch eine didaktische Integration in den Unterrichtsbetrieb werden erörtert. In einem groß angelegten Feldversuch *Traditionelle Lehre versus Multimedia-gestützte Lehre* wurden nicht nur die Motivation, sondern auch der objektive Lernerfolg von Studierenden gemessen, die unterschiedlichen Lernszenarien zugeordnet waren. Dies erlaubt eine objektive positive Bewertung der *Effizienz* der im Rahmen dieser Dissertation entstandenen Lehrmodule.

A few words...

...of acknowledgment usually are placed at this location. And I also wish to express my gratitude to all those who contributed to the formation of this dissertation.

The presented work took shape during my employment as a research assistant in the teleteaching project VIROR and at the Department Praktische Informatik IV, where Prof. Dr. Wolfgang Effelsberg accepted me into his research group on multimedia techniques and computer networks. In this team, I encountered a delightful job surrounding where cooperation, commitment, and freedom of thought were lived and breathed. Prof. Effelsberg not only was my intellectual mentor for this work, he also actively used the teaching modules which were developed during my job title in his lectures. The feedback of the students facilitated their steady improvement. By the way, Prof. Effelsberg was my 'test subject' for both the digital teaching video and the lecture which was stacked up against it for the evaluation introduced in Part III of this work. I am heartily obliged to him for my initiation into the world of science, for tips and clues which have influenced the theme of this work, and for his unfailing support. Prof. Dr. Gabriele Steidl deserves many thanks for having overtaken the co-advising.

I am beholden to my colleagues Stefan Richter, Jürgen Vogel, Martin Mauve, Nicolai Scheele, Jörg Widmer, Volker Hilt, Dirk Farin, and Christian Liebig, as well as to the 'alumni' Werner Geyer and Oliver Schuster for their offers of help in the controversy with my ideas. Be it through precise thematic advice and discussions or through small joint projects which led to common contributions to scientific conferences. Most notably, I want to show my gratitude to Christoph Kuhmünch, Gerald Kühne, and Thomas Haenselmann, who exchanged many ideas with me in form and content and thus facilitated their final transcription. Christoph Kuhmünch and Gert-jan Los sacrificed a share of their week-ends to cross-read my manuscript, to find redundancies and to debug unclear passages. Our system administrator Walter Müller managed the almost flawlessly smooth functioning of the computer systems and our more than unusual secretary Betty Haire Weyerer thoroughly and critically read through my publications in the English language, including the present one, and corrected my 'Genglish', i.e., German-English expressions.

I particularly enjoyed the coaching of 'Studienarbeiten', i.e., students' implementation work, and diploma theses. Among them, I want to name my very first student, Corinna Dietrich, with whom I grew at the task; Holger Wons, Susanne Krabbe, and Christoph Esser signed as contract students at our department after finishing their task — it seems that they had enjoyed it; Sonja Meyer, Timo Müller, Andreas Prassas, Julia Schneider, and Tillmann Schulz helped me to explore different aspects of signal processing, even if not all of their work was related to the presented topic. I owe appreciation to my diploma students Florian Bömers, Uwe Bosecker, Holger Füllner, and Alexander Holzinger for their thorough exploration of and work on facets of the wavelet theory which fit well into the overall picture

of the presented work. They all contributed to my dissertation with their questions and encouragement, with their implementations and suggestions.

The project VIROR permitted me to get in contact with the department Erziehungswissenschaft II of the University of Mannheim. I appreciated this interdisciplinary cooperation especially on a personal level, and it most probably is this climate on a personal niveau which allowed us to cooperate so well scientifically. Here I want to especially thank Holger Horz, and I wish him all the best for his own dissertation project.

In some periods of the formation process of this work, I needed encouraging words more than technical input. Therefore, I want to express my gratitude to my parents, my sister, and my friends for their trust in my abilities and their appeals to my self-assertiveness. My mother, who always reminded me that there is more to life than work, and my father, who exemplified how to question the circumstances and to believe that rules need not always be unchangeable. That the presented work was started, let alone pushed through and completed, is due to Peter Kappelmann, who gives me so much more than a simple life companionship. He makes my life colorful and exciting. This work is dedicated to him.

Claudia Schremmer

Ein paar Worte...

...des Dankes stehen üblicherweise an dieser Stelle. Und auch ich möchte all denen, die mir in irgendeiner Weise bei der Erstellung dieser Arbeit behilflich waren, meine Verbundenheit ausdrücken.

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin in Teleteaching-Projekt VIROR und am Lehrstuhl für Praktische Informatik IV der Universität Mannheim, an den mich Herr Prof. Dr. Wolfgang Effelsberg in seine Forschungsgruppe zu Multi-mediatechnik und Rechnernetzen aufgenommen hat. Dort habe ich ein sehr angenehmes Arbeitsumfeld gefunden, in dem Kooperation, Engagement und geistige Freiheit vorgelebt werden. Er war nicht nur mein geistiger Mentor dieser Arbeit, er hat auch die Lehrmodule, die während meiner Arbeit entstanden, aktiv in der Lehre eingesetzt und es mir dadurch ermöglicht, Rückmeldungen der Studierenden zu berücksichtigen. Ganz nebenbei war Herr Prof. Effelsberg auch meine ‘Versuchsperson’ sowohl für das digitale Lehrvideo als auch für die vergleichende Vorlesung der Evaluation, die in Teil III dieser Arbeit vorgestellt wird. Ich bedanke mich sehr herzlich bei ihm für die Einführung in die Welt der Wissenschaft, für Hinweise und Denkanstöße, die die Thematik dieser Arbeit beeinflussten, und für das Wissen um jeglichen Rückhalt. Frau Prof. Dr. Gabriele Steidl danke ich herzlich für die Übernahme des Korreferats.

Meinen Kollegen Stefan Richter, Jürgen Vogel, Martin Mauve, Nicolai Scheele, Jörg Widmer, Volker Hilt, Dirk Farin und Christian Liebig sowie auch den ‘Ehemaligen’ Werner Geyer und Oliver Schuster danke ich für ihr Entgegenkommen, mir die Auseinandersetzung mit meinen Ideen zu ermöglichen. Vor allem möchte ich mich bedanken bei Christoph Kuhmünch, Gerald Kühne und Thomas Haenselmann, mit denen ich viele inhaltliche Ideen ausgetauscht habe, und die mir das Niederschreiben derselben erleichtert haben. Sei es durch konkrete thematische Ratschläge und Diskussionen oder durch kleine gemeinsame Projekte, die zu gemeinsamen Beiträgen an wissenschaftlichen Konferenzen führten. Christoph Kuhmünch und Gert-Jan Los haben ein gut Teil ihrer Wochenenden geopfert, um mein Manuskript gegenzulesen, Redundanzen zu finden und Unklarheiten zu beseitigen. Unserem Systemadministrator Walter Müller, der sich für das fast immer reibungslose Funktionieren der Systeme verantwortlich zeichnet, und unserer mehr als ungewöhnlichen Sekretärin Betty Haire Weyerer, die mir alle meine englisch-sprachigen Publikationen, inklusive der vorliegenden Arbeit, kritisch durchgesehen hat, gehört an dieser Stelle mein Dank. Selbst wenn die Aussage meiner Sätze nicht geändert wurde, waren die Artikel nach ihrer Durchsicht einfach besser lesbar.

Besonderen Spaß hat mir die Betreuung von Studienarbeiten und Diplomarbeiten gemacht. Dazu zählen: meine erste Studienarbeiterin Corinna Dietrich, mit der zusammen ich an dieser Betreuungsaufgabe gewachsen bin; Holger Wons, Susanne Krabbe und Christoph Esser, die jeweils nach dem Ende ihrer Studienarbeit an unserem Lehrstuhl als ‘HiWi’ gearbeitet haben — es scheint ih-

nen Spaß gemacht zu haben; Sonja Meyer, Timo Müller, Andreas Prassas, Julia Schneider und Tillmann Schulz, die mir geholfen haben, unterschiedliche Aspekte der Signalverarbeitung zu explorieren, selbst wenn nicht alle Arbeiten mit der hier vorgestellten Thematik verbunden waren. Meinen Diplomarbeitern Florian Bömers, Uwe Bosecker, Holger Füßler und Alexander Holzinger gehört ein herzliches Dankeschön für ihre gründliche Einarbeitung in und Aufarbeitung von Teilaspekten der Wavelet Theorie, die zusammen sich in das Gesamtbild der vorliegenden Arbeit fügen. Sie alle haben mit ihren Fragen und Anregungen, mit ihren Programmiertätigkeiten und Vorschlägen zum Gelingen dieser Arbeit beigetragen.

Durch das Projekt VIROR habe ich Kontakt knüpfen dürfen zum Lehrstuhl für Erziehungswissenschaft II der Universität Mannheim. Diese interdisziplinäre Zusammenarbeit hat vor allem auf dem persönlichen Niveau sehr viel Spaß gemacht, und vermutlich war es auch das persönlich gute Klima, das uns hat wissenschaftlich so gut kooperieren lassen. An dieser Stelle spreche ich Holger Horz meinen ausdrücklichen Dank aus und wünsche ihm alles Gute bei seinem eigenen Dissertationsprojekt.

An einigen Punkten in der Entstehungsgeschichte dieser Arbeit habe ich aufmunternde Worte mehr gebraucht als fachlichen Input. Darum möchte ich an dieser Stelle meinen Eltern, meiner Schwester und meinen Freunden Dank sagen für das Zutrauen in meine Fähigkeiten und den Appell an mein Durchsetzungsvermögen. Meine Mutter, die mich stets daran erinnert hat, daß es mehr gibt als Arbeit, mein Vater, der mir als 'Freigeist' vorgelebt hat, Dinge zu hinterfragen und nicht an ein unveränderbares Regelwerk zu glauben. Daß die vorliegende Arbeit aber überhaupt begonnen, geschweige denn durch- und zu Ende geführt wurde, liegt an Peter Kappelmann, der mir so viel mehr gibt als eine einfache Lebensgemeinschaft. Er macht mein Leben bunt und aufregend. Ihm ist diese Arbeit gewidmet.

Claudia Schremmer

Table of Contents

List of Figures	xix
List of Tables	xxii
Notation	xxiii
0 Introduction	1
I Wavelet Theory and Practice	5
1 Wavelets	7
1.1 Introduction	7
1.2 Historic Outline	8
1.3 The Wavelet Transform	9
1.3.1 Definition and Basic Properties	9
1.3.2 Sample Wavelets	10
1.3.3 Integral Wavelet Transform	13
1.3.4 Wavelet Bases	14
1.4 Time–Frequency Resolution	14
1.4.1 Heisenberg’s Uncertainty Principle	14
1.4.2 Properties of the Short–time Fourier Transform	15
1.4.3 Properties of the Wavelet Transform	16

1.5	Sampling Grid of the Wavelet Transform	17
1.6	Multiscale Analysis	18
1.6.1	Approximation	20
1.6.2	Detail	22
1.6.3	Summary and Interpretation	24
1.6.4	Fast Wavelet Transform	26
1.7	Transformation Based on the Haar Wavelet	26
2	Filter Banks	31
2.1	Introduction	31
2.2	Ideal Filters	32
2.2.1	Ideal Low-pass Filter	32
2.2.2	Ideal High-pass Filter	33
2.3	Two-Channel Filter Bank	35
2.4	Design of Analysis and Synthesis Filters	37
2.4.1	Quadrature-Mirror-Filter (QMF)	39
2.4.2	Conjugate-Quadrature-Filter (CQF)	39
3	Practical Considerations for the Use of Wavelets	41
3.1	Introduction	41
3.2	Wavelets in Multiple Dimensions	41
3.2.1	Nonseparability	42
3.2.2	Separability	42
3.3	Signal Boundary	45
3.3.1	Circular Convolution	45
3.3.2	Padding Policies	46
3.3.3	Iteration Behavior	47
3.4	‘Painting’ the Time-scale Domain	47
3.4.1	Normalization	48

3.4.2	Growing Spatial Range with Padding	49
3.5	Representation of ‘Synthesis-in-progress’	50
3.6	Lifting	52
II	Application of Wavelets in Multimedia	57
4	Multimedia Fundamentals	59
4.1	Introduction	59
4.2	Data Compression	60
4.3	Nyquist Sampling Rate	62
5	Digital Audio Denoising	65
5.1	Introduction	65
5.2	Standard Denoising Techniques	66
5.2.1	Noise Detection	67
5.2.2	Noise Removal	67
5.3	Noise Reduction with Wavelets	68
5.3.1	Wavelet Transform of a Noisy Audio Signal	68
5.3.2	Orthogonal Wavelet Transform and Thresholding	69
5.3.3	Nonorthogonal Wavelet Transform and Thresholding	71
5.3.4	Determination of the Threshold	72
5.4	Implementation of a Wavelet-based Audio Denoiser	72
5.4.1	Framework	73
5.4.2	Noise Reduction	74
5.4.3	Empirical Evaluation	77
6	Still Images	81
6.1	Introduction	81
6.2	Wavelet-based Semiautomatic Segmentation	82

6.2.1	Fundamentals	82
6.2.2	A Wavelet-based Algorithm	84
6.2.3	Implementation	86
6.2.4	Experimental Results	86
6.3	Empirical Parameter Evaluation for Image Coding	89
6.3.1	General Setup	89
6.3.2	Boundary Policies	90
6.3.3	Choice of Orthogonal Daubechies Wavelet Filter Bank	93
6.3.4	Decomposition Strategies	94
6.3.5	Conclusion	95
6.3.6	Figures and Tables of Reference	96
6.4	Regions-of-interest Coding in JPEG2000	108
6.4.1	JPEG2000 — The Standard	108
6.4.2	Regions-of-interest	110
6.4.3	Qualitative Remarks	114
7	Hierarchical Video Coding	115
7.1	Introduction	115
7.2	Video Scaling Techniques	116
7.2.1	Temporal Scaling	118
7.2.2	Spatial Scaling	118
7.3	Quality Metrics for Video	119
7.3.1	Vision Models	119
7.3.2	Video Metrics	120
7.4	Empirical Evaluation of Hierarchical Video Coding Schemes	121
7.4.1	Implementation	121
7.4.2	Experimental Setup	122
7.4.3	Results	125

7.4.4	Conclusion	126
7.5	Layered Wavelet Coding Policies	127
7.5.1	Layering Policies	127
7.5.2	Test Setup	129
7.5.3	Results	130
7.5.4	Conclusion	133
7.6	Hierarchical Video Coding with Motion–JPEG2000	134
7.6.1	Implementation	135
7.6.2	Experimental Setup	136
7.6.3	Results	137
7.6.4	Conclusion	138
III	Interactive Learning Tools for Signal Processing Algorithms	141
8	Didactic Concept	143
8.1	Introduction	143
8.2	The Learning Cycle in Distance Education	144
8.2.1	Conceptualization	145
8.2.2	Construction	146
8.2.3	Dialog	146
9	Java Applets Illustrating Mathematical Transformations	147
9.1	Introduction	147
9.2	Still Image Segmentation	148
9.2.1	Technical Basis	148
9.2.2	Learning Goal	149
9.2.3	Implementation	149
9.3	One–dimensional Discrete Cosine Transform	151
9.3.1	Technical Basis	152

9.3.2	Learning Goal	152
9.3.3	Implementation	153
9.4	Two-dimensional Discrete Cosine Transform	155
9.4.1	Technical Basis	155
9.4.2	Learning Goal	155
9.4.3	Implementation	156
9.5	Wavelet Transform: Multiscale Analysis and Convolution	156
9.5.1	Technical Basis	158
9.5.2	Learning Goal	158
9.5.3	Implementation	158
9.6	Wavelet Transform and JPEG2000 on Still Images	160
9.6.1	Technical Basis	160
9.6.2	Learning Goal	160
9.6.3	Implementation	161
9.6.4	Feedback	163
10	Empirical Evaluation of <i>Interactive Media in Teaching</i>	165
10.1	Introduction	165
10.2	Test Setup	166
10.2.1	Learning Setting	166
10.2.2	Hypotheses	168
10.3	Results	169
10.3.1	Descriptive Statistics	170
10.3.2	Analysis of Variance	172
11	Conclusion and Outlook	179

IV	Appendix	181
A	Original Documents of the Evaluation	183
A.1	Computer-based Learning Setting	183
A.1.1	Setting: <i>Exploration</i>	184
A.1.2	Setting: <i>Script</i>	185
A.1.3	Setting: β -Version	188
A.1.4	Setting: $c't$ -Article	189
A.2	Knowledge Tests	191
A.2.1	Preliminary Test	191
A.2.2	Follow-up Test	193
A.2.3	Sample Solutions	198
A.3	Quotations of the Students	200

List of Figures

1.1	Sample wavelets	12
1.2	The Mexican hat wavelet and two of its dilates and translates, including the normalization factor	13
1.3	Time–frequency resolution of the short–time Fourier transform and the wavelet transform	16
1.4	Sampling grids of the short–time Fourier and the dyadic wavelet transforms	18
1.5	Multiscale analysis	19
1.6	Scaling equation: heuristic for the indicator function and the <i>hat</i> function	21
1.7	Subband coding	25
1.8	Tiling the time–scale domain for the dyadic wavelet transform	26
1.9	Haar transform of a one–dimensional discrete signal	28
2.1	Ideal low–pass and high–pass filters	34
2.2	Two–channel filter bank	36
2.3	Arbitrary low–pass and high–pass filters	36
3.1	Separable wavelet transform in two dimensions	44
3.2	Circular convolution versus mirror padding	46
3.3	Two possible realizations of ‘painting the time–scale coefficients’	48
3.4	Trimming the approximation by zero padding and mirror padding	50
3.5	Representation of synthesis–in–progress	51
3.6	Analysis filter bank for the fast wavelet transform with lifting	52
3.7	Lifting scheme: prediction for the odd coefficients	53

3.8	The lifting scheme	54
4.1	Digital signal processing system	59
4.2	Hybrid coding for compression	61
5.1	Effect of wavelet-based thresholding of a noisy audio signal	70
5.2	Hard and soft thresholding, and shrinkage	71
5.3	Graphical user interface of the wavelet-based audio tool	74
5.4	Selected features of the wavelet-based digital audio processor	75
5.5	Visualizations of the time-scale domain and of the time domain	76
5.6	Visible results of the denoising process	78
6.1	<i>Pintos</i> by Bev Doolittle	83
6.2	In the search for a next rectangle, a ‘candidate’ is rotated along the ending point . . .	85
6.3	Example for semiautomatic segmentation	86
6.4	Test images for the empirical evaluation of the different segmentation algorithms . .	87
6.5	Impact of different wavelet filter banks on visual perception	94
6.6	Impact of different decomposition strategies on visual perception	95
6.7	Test images for the empirical parameter evaluation	97
6.8	Test images with threshold $\lambda = 10$ in the time-scale domain	98
6.9	Test images with threshold $\lambda = 20$ in the time-scale domain	99
6.10	Test images with threshold $\lambda = 45$ in the time-scale domain	100
6.11	Test images with threshold $\lambda = 85$ in the time-scale domain	101
6.12	Average visual quality of the test images at the quantization thresholds $\lambda =$ 10, 20, 45, 85	104
6.13	Average bit rate heuristic of the test images at the quantization thresholds $\lambda =$ 10, 20, 45, 85	105
6.14	Mean visual quality of the test images at the quantization thresholds $\lambda = 10, 20, 45, 85$ with standard versus nonstandard decomposition	107
6.15	Classification according to image content	111
6.16	Classification according to visual perception of distance	112

6.17	Two examples of a pre-defined shape of a region-of-interest	112
6.18	Region-of-interest mask with three quality levels	113
7.1	Layered data transmission in a heterogeneous network	116
7.2	Temporal scaling of a video stream	118
7.3	Visual aspect of the artifacts of different hierarchical coding schemes	124
7.4	Layering policies of a wavelet-transformed image with decomposition depth 3 . . .	128
7.5	Frame 21 of the test sequence <i>Traffic</i> , decoded with the layering policy 2 at 6.25% of the information	129
7.6	Average PSNR value of the Table 7.4 for different percentages of synthesized wavelet coefficients	131
7.7	Frame 21 of the test sequence <i>Traffic</i>	132
7.8	Linear sampling order of the coefficients in the time-scale domain	133
7.9	Sampling orders used by the encoder before run-length encoding	135
7.10	GUI of our motion-JPEG2000 video client	136
8.1	Learning cycle	145
9.1	Graphical user interface of the segmentation applet	150
9.2	Effects of smoothing an image and of the application of different edge detectors . . .	151
9.3	DCT: Subsequent approximation of the sample points by adding up the weighted frequencies.	153
9.4	GUI of the DCT applet	154
9.5	Examples of two-dimensional cosine basis frequencies	156
9.6	GUI of the 2D-DCT applet	157
9.7	Applet on multiscale analysis and on convolution-based filtering	159
9.8	Different display modes for the time-scale coefficients	161
9.9	The two windows of the wavelet transform applet used on still images	162
10.1	Photos of the evaluation of the computer-based learning setting	167
A.1	c't-Article	190

List of Tables

1.1	Relations between signals and spaces in multiscale analysis	24
3.1	The number of possible iterations on the approximation part depends on the selected wavelet filter bank	47
3.2	The size of the time–scale domain with padding depends on the selected wavelet filter bank	49
3.3	Filter coefficients of the two default wavelet filter banks of JPEG2000	55
4.1	Classification of compression algorithms	62
5.1	Evaluation of the wavelet denoiser for <code>dnbloop.wav</code>	79
6.1	Experimental results for three different segmentation algorithms	88
6.2	Experimental results: summary of the four test images	88
6.3	Detailed results of the quality evaluation with the PSNR for the six test images . . .	102
6.4	Heuristic for the compression rate of the coding parameters of Table 6.3	103
6.5	Average quality of the six test images	104
6.6	Average bit rate heuristic of the six test images	105
6.7	Detailed results of the quality evaluation for the standard versus the nonstandard decomposition strategy	106
6.8	Average quality of the six test images in the comparison of standard versus nonstandard decomposition	107
6.9	Structure of the JPEG2000 standard	108
7.1	Test sequences for hypothesis $H_{1,0}$	125

7.2	Correlation between the human visual perception and the PSNR, respectively the DIST metric and its sub-parts	125
7.3	Evaluation of the four layered video coding schemes	126
7.4	The PSNR of frame 21 of the test sequence <i>Traffic</i> for different decoding policies and different percentages of restored information	130
7.5	Heuristics for the bit rate of a wavelet encoder for frame 21 of the test sequence <i>Traffic</i> with different wavelet filters	134
7.6	Results of the performance evaluation for a 64 kbit/s ISDN line	138
7.7	Results of the performance evaluation for a 10 Mbit/s LAN connection	139
10.1	Descriptive statistics on the probands	170
10.2	Descriptive statistics on the probands, detailed for the setting	171
10.3	Test of the significance and explained variance of inter-cell dependencies for hypothesis $H_{1;0}$	173
10.4	Estimated mean values, standard deviation and confidence intervals of the dependent variable at the different learning settings for hypothesis $H_{1;0}$	174
10.5	Test of the significance and explained variance of inter-cell dependencies for hypothesis $H_{2;0}$	176
10.6	Estimated mean values, standard deviation and confidence intervals of the dependent variable at the different learning settings for hypothesis $H_{2;0}$	177

Notation

Sets

\mathbb{Z}	Integers
\mathbb{R}	Real numbers
\mathbb{C}	Complex numbers
$L_1(\mathbb{R})$	Banach space of all absolute integrable functions: $\int_{\mathbb{R}} f(t) dt < \infty$
$L_2(\mathbb{R})$	Hilbert space of all square integrable functions: $\int_{\mathbb{R}} f(t) ^2 dt < \infty$
l_2	Set of sequences $\{(a_k)_{k \in \mathbb{Z}}\}$ such that $\sum_{k \in \mathbb{Z}} a_k ^2 < \infty$
V_{2j}	Approximation space in multiscale approximation
W_{2j}	Detail space in multiscale analysis
$U \oplus V$	Direct sum of two vector spaces U and V
$U \times V$	Tensor product of two vector spaces U and V

Symbols

$ \cdot $	Absolute value
z^*	Complex conjugate of $z \in \mathbb{C}$
f^*	Complex conjugate of the complex function f
$\ \cdot\ $	Norm in $L_2(\mathbb{R})$: $\ f\ = \int_{\mathbb{R}} f(t) ^2 dt$
$\langle \cdot, \cdot \rangle$	Inner product in $L_2(\mathbb{R})$: $\langle f, g \rangle = \int_{\mathbb{R}} f(t) g^*(t) dt$
t	Variable in time domain
ω	Variable in frequency domain
$[a, b]$	Closed interval from a to b
$]a, b[$	Open interval from a to b
$[a, b[$	Interval including a and excluding b
σ	Variance of a random variable
\bar{x}	Mean value of a random variable
η	Explained variance
E	Expectation of a random variable
I	Identity matrix
det	Determinant of a matrix
\gg	Much bigger

Signals

$f(t)$	Continuous time signal
$f[k]$	Coefficients in Fourier series
$f * g$	Convolution of f and g
$\delta_{k,n}$	$\delta_{k,n} = 1$ if $k = n$ and 0 else
$\mathbf{1}_{[a,b[}$	Indicator function on the interval $[a, b[$
ψ	Wavelet
$\psi_{a,b}$	weighted dilated and shifted wavelet: $\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$
φ	Scaling function
h_0	Filter mask for scaling function φ
h_1	Filter mask for wavelet ψ
H_0	System function to h_0
H_1	System function to h_1

Transforms

\hat{f}	Fourier transform of f : $\hat{f}(\omega) = \langle f, e^{2i\pi t\omega} \rangle = \int_{\mathbb{R}} f(t)e^{-2i\pi t\omega} dt$
\tilde{f}_ψ	Wavelet transform of f with respect to ψ : $\tilde{f}_\psi = \langle f, \psi \rangle = \int_{\mathbb{R}} f(t)\psi^* dt$
$\mathcal{A}_{f,k}^j$	Approximation of f at the scale 2^j
$\mathcal{D}_{f,k}^j$	Detail of f at the scale 2^j

Misc

DCT	Discrete cosine transform
WT	Wavelet transform
Hz	Hertz, i.e, quantity per second
DC	Direct currency
AC	Alternating currency
fps	frames per second
dB	decibel
HVP	Human visual perception
HVS	Human visual system
JPEG	Joint photographic experts group
ISO	International standardizations organization
ITU	International telecommunications union
ITS	Institute for telecommunication sciences
ROI	Region-of-interest
RHQ	Region of higher quality
RMQ	Region of minor quality

Chapter 0

Introduction

*Wanting is not enough; desiring only makes
you reach the target.*

– Ovid

Motivation

In recent years, the processing of multimedia data streams such as audio, images, and digital video has experienced a rapidly expanding distribution. In Germany, the popular use of the Internet is fueling a steadily increasing demand for multimedia content. Given the availability of adequate computing performance and storage capacity, a steadily growing amount of multimedia data are routinely digitally transferred — and in many a newspaper, the trained eye recognizes the compression algorithm underlying the artifacts of a printed title photo. But also time-critical applications like audio or video are a source of interest to many users who download the data from the Internet and play them back on a multimedia-PC equipped with microphone and loudspeakers. This demand — and with it the supply — have increased at a much faster pace than hardware improvements. Thus, there is still a great need for efficient algorithms to compress and efficiently transmit multimedia data.

The wavelet transform renders an especially useful service in this regard. It decomposes a signal into a multiscale representation and hence permits a precise view of its information content. This can be successfully exploited for two different, yet related purposes:

- *Content Analysis.* Content analysis of multimedia data seeks to semantically interpret digital data. In surveying an audio stream for example, it aims to automatically distinguish *speech* from *music*. Or an interesting *object* could be extracted from a video sequence. Content analysis most often is a pre-processing step for a subsequent algorithm. An audio equalizer, for instance, needs information about which data of an audio stream describe what frequency bands before it can reinforce or attenuate them specifically. A human viewer of a digital image or video will have fewer objections to a background coded in lower quality as long as the actual object of interest is displayed in the best possible quality.

- *Compression.* Compression demands efficient coding schemes to keep the data stream of a digital medium as compact as possible. This is achieved through a re-arrangement of the data (i.e., *lossless* compression) as well as through truncation of part of the data (i.e., *lossy* compression). Lossy algorithms make clever use of the weaknesses of human auditory and visual perception to first discard information that humans are not able to perceive. For instance, research generally agrees that a range from 20 Hz to 20 kHz is audible to humans. Frequencies outside this spectrum can be discarded without perceptible degradation.

This is where the research on a representation of digital data enters that best mirrors human perception. Due to its property of preserving both time, respectively, location, and frequency information of a transformed signal, the wavelet transform renders good services. Furthermore, the ‘zooming’ property of the wavelet transform shifts the focus of attention to different scales. Wavelet applications encompass audio analysis, and compression of still images and video streams, as well as the analysis of medical and military signals, methods to solve boundary problems in differential equations, and regularization of inverse problems.

In opposition to hitherto common methods of signal analysis and compression, such as the short-time Fourier and cosine transforms, the wavelet transform offers the convenience of less complexity. Furthermore, rather than denoting a specific function, the term *wavelet* denotes a class of functions. This has the drawback that a specific function still has to be selected for the transformation process. At the same time, it offers the advantage to select a transformation-wavelet according to both the signal under consideration and the purpose of the transformation, and thus to achieve better results.

We will show that the wavelet transform is especially suited to restore a noisy audio signal: The uncorrelated noise within a signal remains uncorrelated, thus thresholding techniques allow detection and removal of the noise. Our prototype implementation of a wavelet-based audio denoiser allows various parameters to be set flexibly. We hereby underline the practical potential of the theoretical discussion.

The multiscale property of the wavelet transform allows us to track a predominant structure of a signal in the various scales. We will make use of this observation to develop a wavelet-based algorithm for the semiautomatic edge detection in still images. Hence, we will show that the wavelet transform allows a *semantic* interpretation of an image. Various evaluations on the setting of the parameters for the wavelet transform on still images finally will allow us to recommend specific settings for the boundary, the filter bank, and the decomposition of a still image.

In the Internet, many users with connections of different bandwidths might wish to access the same video stream. In order to prevent the server from stocking multiple copies of a video at various quality levels, hierarchical coding schemes are sought. We will successfully use the wavelet transform for hierarchical video coding algorithms. This novel approach to the distribution of the transformed coefficients onto different quality levels of the encoded video stream allows various policies. Empirical evaluations of a prototype implementation of a hierarchical video server and a corresponding client indicate that wavelet-based hierarchical video encoding is indeed a promising approach.

Outline

This dissertation is divided into three major parts. The first part reviews the theory of wavelets and the dyadic wavelet transform and thus provides a mathematical foundation for the following. The second part presents our contributions to novel uses of the wavelet transform for the coding of audio, still images, and video. The final part addresses the teaching aspect with regard to students in their first semesters of study, where we propose new approaches to multimedia-enhanced teaching.

Chapter 1 reviews the fundamentals of the wavelet theory: We discuss the time–frequency resolution of the wavelet transform and compare it to the common short–time Fourier transform. The multiscale property of the dyadic wavelet transform forms the basis for our further research on multimedia applications; it is introduced, explained, and visualized in many different, yet each time enhanced, tableaux. An example of the Haar transform aims to render intuitive the idea of low–pass and high–pass filtering of a signal before we discuss the general theoretical foundation of filter banks in Chapter 2. Practical considerations for the use of wavelets in multimedia are discussed in Chapter 3. We focus on the convolution–based implementation of the wavelet transform since we consider the discussion of all these parameters important for a substantial understanding of the wavelet transform. Yet the implementation of the new image coding standard JPEG2000 with its two suggested standard filters is outlined.

After a brief introduction into the fundamentals of multimedia coding in Chapter 4, Chapter 5 presents the theory of wavelet–based audio denoising. Furthermore, we present our implementation of a wavelet–based audio denoising tool. Extending the wavelet transform into the second dimension, we suggest a novel, wavelet–based algorithm for semiautomatic image segmentation and evaluate the best parameter settings for the wavelet transform on still images in Chapter 6. A critical discussion of the region–of–interest coding of JPEG2000 concludes the investigation of still images. Chapter 7 contains our major contribution: the application of the wavelet transform to hierarchical video coding. We discuss this novel approach to successfully exploit the wavelet transform for the distribution of the transformed and quantized coefficients onto different video layers, and present a prototype of a hierarchical client–server video application.

In our daily work with students in their first semesters of study, we encountered many didactic shortcomings in the traditional teaching of mathematical transformations. After an introduction into our didactic concept to resolve this problem in Chapter 8, we present a number of teachware programs in Chapter 9. Chapter 10 presents an evaluation on the learning behavior of students with new multimedia-enhanced tools. In this survey, we evaluate the learning progress of students in a ‘traditional’ setting with a lecture hall and a professor against that of students in a computer–based scenario and show that the success and the failure of multimedia learning programs depend on the precise setting.

Chapter 11 concludes this dissertation and looks out onto open questions and future projects.

Part I

Wavelet Theory and Practice

Chapter 1

Wavelets

My dream is to solve problems, with or without wavelets.

– Bruno Torresani

1.1 Introduction

This chapter introduces the concept of the wavelet transform on digital signals. The wavelet transform carries out a special form of analysis by shifting the original signal from the time domain into the time–frequency, or, in this context, *time–scale* domain. The idea behind the wavelet transform is the definition of a set of basis functions that allow an *efficient, informative* and *useful* representation of signals. Having emerged from an advancement in time–frequency localization from the short–time Fourier analysis, the wavelet theory provides facilities for a flexible analysis as wavelets figuratively ‘zoom’ into a frequency range. Wavelet methods constitute the underpinning of a new comprehension of time–frequency analysis. They have emerged independently within different scientific branches of study until all these different viewpoints have been subsumed under the common terms of *wavelets* and *time–scale analysis*. The contents of this first part of the dissertation were presented in a tutorial at the International Symposium on Signal Processing and Its Applications 2001 [Sch01d].

A historic overview of the development of the wavelet theory precedes the introduction of the (one–dimensional) continuous wavelet transform. Here, the definition of a wavelet and basic properties are given and sample wavelets illustrate the concepts of these functions. After defining the integral wavelet transform, we review the fact that a particular sub–class of wavelets that meet our requirements forms a basis for the space of square integrable functions. In the section about time–frequency resolution, a mathematical foundation is presented, and it is shown why wavelets ‘automatically’ adapt to an interesting range in frequency resolution and why their properties — depending on the application — might be superior to the short–time Fourier transform. The design of multiscale analysis finally leads directly to what is commonly referred to as the *fast wavelet transform*. The example of a transformation based on the Haar wavelet concludes this introductory chapter. Chapter 2 reviews the general design of analysis and synthesis filter banks for a multiscale analysis. This mathematical survey puts the construction of wavelet filter banks into a general context and illustrates the conjugate–quadrature wavelet filters used during our evaluations in Part II. The explanations in Chapters 1 and

2 are inspired by [Mal98] [LMR98] [Ste00] [Dau92] [Boc98], and [Hub98]. Chapter 3 presents our own contribution to the discussion of practical considerations for the use of wavelets. The topics assessed include the discussion of wavelet filter banks in multiple dimensions, different policies to handle signal boundaries, the challenge to represent the coefficients in the wavelet-transformed time-scale domain, and policies to represent a decoded signal when the decoder has not yet received the complete information due to network delay or similar reasons.

1.2 Historic Outline

The wavelet theory combines developments in the scientific disciplines of pure and applied mathematics, physics, computer science, informatics, and engineering. Some of the approaches date back until the early beginning of the 20th century (e.g., Haar wavelet, 1910). Most of the work was done around the 1930s, though at that time, the separate efforts did not appear to be parts of a coherent theory. Daubechies compares the history of wavelets to a tree with many roots growing in distinct directions. The trunk of the tree denotes the joint forces of scientists from different branches of study in the development of a *wavelet theory*. The branches are the different directions and applications which incorporate wavelet methods.

One of the wavelet roots was put down around 1981 by Morlet [MAFG82] [GGM85]. At that time, the standard tool for time-frequency analysis was the short-time Fourier transform. However, as the size of the analyzing window is fixed, it has the disadvantage of being imprecise about time at high frequencies unless the analyzing window is downsized, which means that information about low frequencies is lost. In his studies about how to discover underground oil, Morlet varied the concept of the transform. Instead of keeping the size of the window fixed and filling it with oscillations of different frequencies, he tried the reverse: He kept the number of oscillations within the window constant and varied the width of the window. Thus, Morlet obtained a good time resolution of high frequencies and simultaneously a good frequency resolution of low frequencies. He named his functions *wavelets of constant shape*.

The theoretical physicist Grossmann proved that the discrete, and critically sampled wavelet transform was reversible, thus no error was introduced by transform and inverse transform, i.e., analysis and synthesis [GM85] [GMP85].

In 1985, the mathematician Meyer heard of the work of Morlet and Grossmann. He was convinced that, unlike the dyadic approach of Morlet and Grossmann, a good time-frequency analysis requires redundancy [Mey92] [Mey93] [Mey87]. This continuous wavelet transform inspired other approaches. As far as the continuous transform is concerned, nearly any function can be called a wavelet as long as it has a vanishing integral. This is not the case for (nontrivial) orthogonal wavelets. In an attempt to prove that such orthogonal wavelets do not exist, Meyer ended up doing exactly the opposite, and constructing precisely the kind of wavelet he thought didn't exist. [Hub98]

In 1986 Mallat, who worked in image analysis and computer vision, became preoccupied with the new transform. He was familiar with scale-dependent representations of images, among others due to the principle of the Laplace pyramid of Burt and Adelson [BA83]. Mallat and Meyer realized that the multiresolution with wavelets was a different version of an approach long been applied by electrical engineers and image processors. They managed to associate the wavelet transform to the

multiscale analysis and to calculate the transform filters recursively. The idea to not extract the filter coefficients from the wavelet basis, but conversely, to use a filter bank to construct a wavelet basis led to a first wavelet basis with compact support in 1987 [Mal87]. Mallat also introduced the notion of a *scaling function* — which takes the counterpart of the wavelets — into his work, and proved that multiresolution analysis is identical to the discrete fast wavelet transform. [Boc98]

While Mallat first worked on truncated versions of infinite wavelets, Daubechies [Dau92] introduced a new kind of orthogonal wavelet with *compact support*. This new class of wavelets made it possible to avoid the errors caused by truncation. The so-called Daubechies wavelets have no closed representation; they are constructed via iterations. In addition to orthogonality and compact support, Daubechies was seeking smooth wavelets with a high order of vanishing moments¹. Daubechies wavelets provide the smallest support for the given number of vanishing moments [Dau92]. In 1989, Coifman suggested to Daubechies that it might be worthwhile to construct orthogonal wavelet bases with vanishing moments not only for the wavelet, but also for the scaling function. Daubechies constructed the resulting wavelets in 1993 [Dau92] and named them *coiflets*.

Around this time, wavelet analysis evolved from a mathematical curiosity to a major source of new signal processing algorithms. The subject branched out to construct wavelet bases with very specific properties, including orthogonal and biorthogonal wavelets, compactly supported, periodic or interpolating wavelets, separable and nonseparable wavelets for multiple dimensions, multiwavelets, and wavelet packets. [Wic98] [Ste00]

1.3 The Wavelet Transform

The aim of signal processing is to extract specific information from a given function f which we call a *signal*. For this purpose, there is mainly one idea: to transform the signal in the expectation that a well-suited transformation will facilitate the reading, i.e., the *analysis* of the relevant information. Of course, the choice of the transform depends on the nature of the information one is interested in. A second demand on the transform is that the original function can be *synthesized*, i.e., reconstructed from its transformed state. This is the claim for invertibility.

This section investigates the definition and nature of wavelets. The continuous wavelet transform is presented and its most important features are discussed.

1.3.1 Definition and Basic Properties

Definition 1.1 A wavelet is a function $\psi \in L_2(\mathbb{R})$ which meets the admissibility condition

$$0 < c_\psi := 2\pi \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \quad (1.1)$$

where $\hat{\psi}$ denotes the Fourier transform of the wavelet ψ .

¹Vanishing moments are explained in Section 1.3.

The constant c_ψ designates the admissibility constant [LMR98]. Approaching $\omega \rightarrow 0$ gets critical. To guarantee that Equation (1.1) is accomplished, we must ensure that $\hat{\psi}(0) = 0$. It follows that a wavelet integrates to zero:

$$0 = \hat{\psi}(0) = \int_{\mathbb{R}} \psi(t) e^{-2i\pi t \cdot 0} dt = \int_{\mathbb{R}} \psi(t) dt. \quad (1.2)$$

Thus, a wavelet has the same volume ‘above the x-axis’ as ‘below the x-axis’. This is where the name wavelet, i.e., little wave, originates.

Since $\psi \in L_2(\mathbb{R})$, also is its Fourier transform $\hat{\psi} \in L_2(\mathbb{R})$: $\int_{\mathbb{R}} |\hat{\psi}(\omega)|^2 d\omega < \infty$. Therefore, $|\hat{\psi}(\omega)|$ declines sufficiently fast for $|\omega| \gg 0$. In practical considerations, it is sufficient that the majority of the wavelet’s energy is restricted to a finite interval. This means that a wavelet has strong localization in the time domain.

1.3.2 Sample Wavelets

The definition of a wavelet is so general that a ‘wavelet’ can have very different properties and shapes. As we will see later in this chapter, multiscale analysis links wavelets to high-pass filters, respectively, band-pass filters. The theory of filter banks is detailed in Section 2.3. In the following, we present some of the most common wavelets and their Fourier transforms.

1.3.2.1 Haar Wavelet

Long before engineers and mathematicians began to develop the wavelet theory, Haar [Haa10] had made use of the following function:

$$\psi(t) = \begin{cases} 1 & : 0 \leq t < \frac{1}{2}, \\ -1 & : \frac{1}{2} \leq t \leq 1, \\ 0 & : \text{else.} \end{cases}$$

The Haar wavelet is demonstrated in Figure 1.1 (a). Its Fourier transform is

$$\hat{\psi}(\omega) = \frac{1}{2i\pi\omega} \left(1 - e^{-i\pi\omega}\right)^2 = i \sin(\pi\omega/2) \text{sinc}(\pi\omega/2) e^{-i\pi\omega}.$$

where the sinc function is defined as $\text{sinc}(x) = \frac{\sin(x)}{x}$. This means, that $|\hat{\psi}|$ is an even function.

1.3.2.2 Mexican Hat Wavelet

The Mexican Hat wavelet is an important representative of the general theorem that if a function $f \in L_1(\mathbb{R})$ is a continuously differentiable function and its derivative $\psi = f' \in L_2(\mathbb{R})$, then ψ

accomplishes the admissibility condition (1.1) [LMR98]. The Mexican Hat owes its name to its shape (see Figure 1.1 (b)). It is defined as the second derivative of a Gaussian [Mur88],

$$\psi(t) = -\frac{d^2}{dt^2}e^{-t^2/2} = (1 - t^2)e^{-t^2/2}.$$

Its Fourier transform is $\hat{\psi}(\omega) = 4\pi^2\omega^2\sqrt{2\pi}e^{-2\pi^2\omega^2}$.

1.3.2.3 Morlet Wavelet

The lower bound of time–frequency resolution (see Section 1.4) is reached by the Morlet wavelet [GGM85]. It is a modulated Gaussian, adjusted slightly so that $\hat{\psi}(0) = 0$, with the Fourier transform

$$\hat{\psi}(\omega) = \pi^{-\frac{1}{4}} \left[e^{-(\omega-\omega_0)^2/2} - e^{-\omega^2/2} e^{-\omega_0^2/2} \right]. \quad (1.3)$$

The wavelet thus has the form

$$\psi(t) = \sqrt{2}\pi^{\frac{1}{4}} \left[e^{-i2\pi\omega_0 t} - e^{-\omega_0^2/2} \right] e^{-2\pi^2 t^2},$$

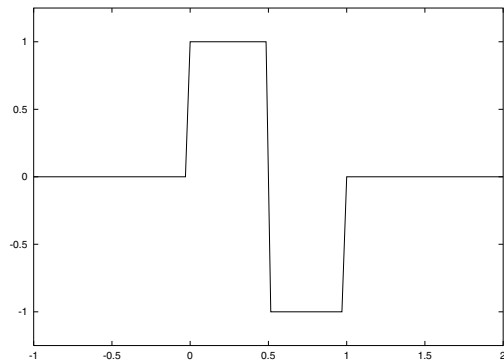
where ω_0 is a constant that often is chosen such that the ratio of the highest and the second highest maximum of ψ is sufficiently large. In practice, one often sets $\omega_0 = 5$. For this value of ω_0 , the second term in Equation (1.3) is so small that it can be neglected in practice [Dau92]. The shape of the real part of the Morlet wavelet is demonstrated in Figure 1.1 (c).

1.3.2.4 Daubechies Wavelet

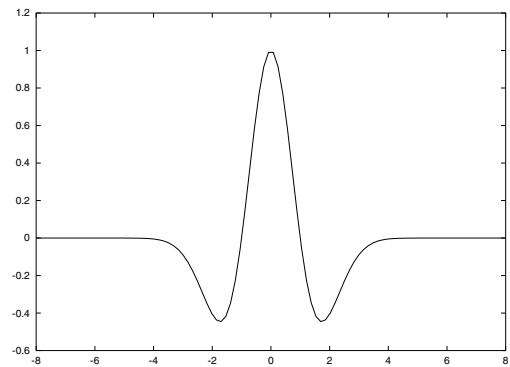
The family of Daubechies wavelets is most often used for multimedia implementations. They are a specific occurrence of the conjugate-quadrature filters (see Section 2.4.2), whose general theory is outlined in Chapter 2.

The Daubechies wavelets (see Figure 1.1 (d)) are obtained by iteration; no closed representation exists. The Daubechies wavelets are the shortest compactly supported orthogonal wavelets for a given number of vanishing moments² [Dau92]. The degree n_0 of vanishing moments determines the amount of filter bank coefficients to $2n_0$.

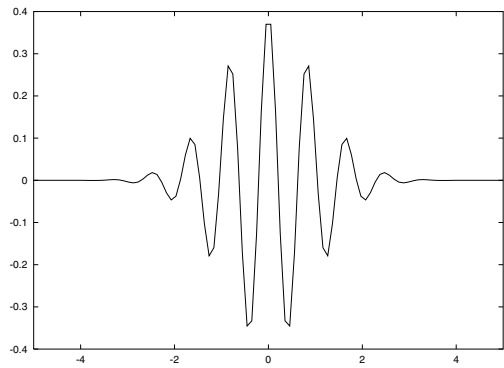
²A function f has n_0 vanishing moments, if for $p = 0, \dots, n_0 - 1$ applies: $\int_{\mathbb{R}} t^p f(t) dt = 0$. If ψ has enough vanishing moments, then the wavelet coefficients $\tilde{f}_{\psi}(a, b)$ (see Equation (1.4)) are small at fine scales $a = 2^j$ (see also Section 1.6). This is a desirable property for compression.



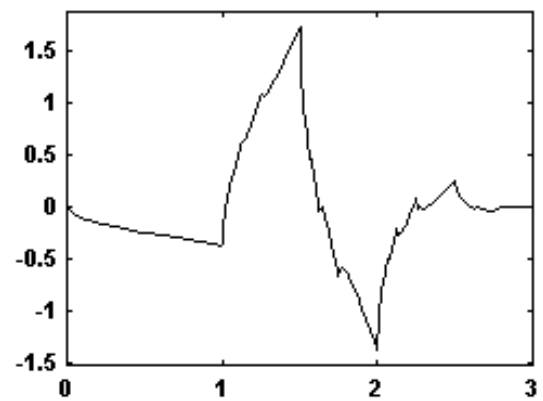
(a) Haar wavelet.



(b) Mexican Hat.



(c) Real part of Morlet wavelet.



(d) Daubechies-2 wavelet.

Figure 1.1: Sample wavelets.

1.3.3 Integral Wavelet Transform

Definition 1.2 The integral wavelet transform of a function $f \in L_2(\mathbb{R})$ with regard to the admissible wavelet ψ is given by

$$f \mapsto \tilde{f}_\psi(a, b) := \frac{1}{\sqrt{a}} \int_{\mathbb{R}} f(t) \psi^* \left(\frac{t-b}{a} \right) dt = \int_{\mathbb{R}} f(t) \psi_{a,b}^*(t) dt, \quad (1.4)$$

where ψ^* is the complex conjugate of ψ . The scalar $a > 0$ is the dilation or scale factor, b is the translation parameter, and the factor $\frac{1}{\sqrt{a}}$ enters Equation (1.4) for energy normalization across the different scales (see Section 1.4.1), thus $\psi_{a,b}$ denotes a weighted dilated and translated wavelet.

To illustrate Equation (1.4), we detail the effects of a compactly supported wavelet ψ . The translation parameter b shifts the wavelet so that $\tilde{f}_\psi(a, b)$ contains local information of f at time $t = b$. The parameter a manages the area of influence: With $a \rightarrow 0$, the wavelet transform ‘zooms’ into the location $t = b$ while $a \gg 0$ blurs the time-resolution. Figure 1.2 demonstrates the idea behind dilation and translation.

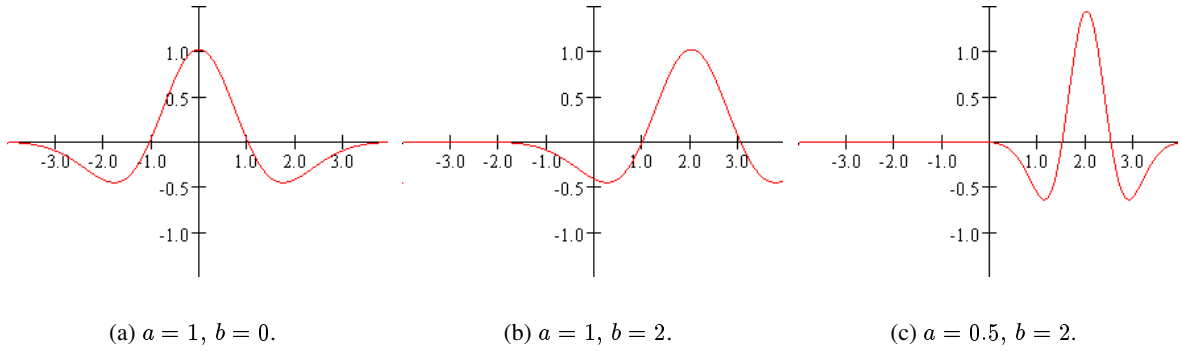


Figure 1.2: The Mexican hat wavelet and two of its dilates and translates, including the normalization factor.

The wavelet transform of a signal f examines the signal with the help of the wavelet ψ . In other words, one builds L_2 -scalar products of f and $\psi_{a,b}$, which denotes the dilated and translated versions of ψ . It is important to note that no wavelet basis has yet been specified. The theory of wavelet transforms relies on general properties of the wavelets. It is a framework within which one can define wavelets according to the requirements.

Observation 1.1 The wavelet transform as defined in Equation (1.4) is linear:

$$\begin{aligned} \lambda f &\mapsto \widetilde{\lambda f}_\psi(a, b) = \lambda \tilde{f}_\psi(a, b), \\ f + g &\mapsto \widetilde{(f + g)}_\psi(a, b) = \tilde{f}_\psi(a, b) + \tilde{g}_\psi(a, b). \end{aligned}$$

1.3.4 Wavelet Bases

A wavelet transform decomposes a signal f into coefficients for a corresponding wavelet ψ . As all wavelets ‘live’ in $L_2(\mathbb{R})$, we would like to know whether *every* function $f \in L_2(\mathbb{R})$ can be approximated with arbitrary precision. This is the case: The set of wavelets

$$\Psi = \{\psi \in L_2(\mathbb{R}) : \psi \text{ is admissible}\}$$

is a *dense* subset of $L_2(\mathbb{R})$. That is, every function in L_2 can be approximated by wavelets, and the approximation error shrinks arbitrarily. [LMR98]

Moreover, we can demand that the wavelet basis have a special appearance: The set $\{\psi_{a,b} : (a,b) \in \Lambda\}$ is a wavelet basis, where ψ denotes an admissible wavelet, Λ denotes an arbitrary set of indices and $\psi_{a,b}$ signifies the dilated and translated versions of the wavelet ψ . Thus, we can approximate all functions $f \in L_2(\mathbb{R})$ by a set of wavelet coefficients $\{c_{a,b} : (a,b) \in \Lambda\}$ [Boc98]:

$$f = \sum_{(a,b) \in \Lambda} c_{a,b} \psi_{a,b}. \quad (1.5)$$

Equation (1.5) says that every square integrable function can be approximated by dilated and translated versions of one wavelet only. This is the reason why our considerations focus on this class of wavelets. In Section 1.5, we will further see that the dilation and translation parameters can be strongly restricted, while still maintaining the property that no information gets lost and the wavelet transform is reversible. This leads to the fast wavelet transform.

1.4 Time–Frequency Resolution

In the previous sections we have mentioned that the wavelet transform decomposes a one–dimensional signal into the two dimensions of time and frequency. We have further shown that a wavelet ‘zooms’ into a selected frequency. This section elaborates the background of this zooming property. A comparison to the short–time Fourier transform illustrates the common properties, but also the differences between the two approaches.

1.4.1 Heisenberg’s Uncertainty Principle

It would be a nice feature of time–frequency analysis of a signal f if a signal whose energy is well localized in time could have a Fourier transform whose energy is well concentrated in a small frequency neighborhood. In order to reduce the time–spread of f , a scaling factor a is introduced. If we denote the total energy of the signal by $\|\cdot\|$ (i.e., the canonical norm in $L_2(\mathbb{R})$), and aim to keep the total energy of both f and the time–scaled signal f_a the same, i.e., $\|f\|^2 = \|f_a\|^2$, then it follows that

$$f_a(t) = \frac{1}{\sqrt{a}} f\left(\frac{t}{a}\right).$$

Regarding the Fourier transform of the time–scaled signal f_a , we get

$$\hat{f}_a(\omega) = \sqrt{a} \hat{f}(a\omega).$$

This means that the amount of localization gained in time by dividing the time instant by a is lost in frequency resolution. The underlying principle is the trade–off between time and frequency localization. This principle was discovered and proven by Heisenberg during his studies on quantum mechanics [Mes61]. Moreover, a lower bound for the reachable precision exists. If σ_t denotes the time–spread around a center instant, and σ_ω denotes the frequency spread around a center frequency, the *Heisenberg Uncertainty Principle* states that

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4}.$$

The boxes $\sigma_t \sigma_\omega$ are commonly known as *Heisenberg boxes*. They define the total time–frequency uncertainty and allow a graphic interpretation.

1.4.2 Properties of the Short–time Fourier Transform

For the short–time Fourier analysis the signal f is multiplied by a real and symmetric window g before the integral Fourier transform decomposes the signal into its frequencies. The window g is translated by ν and modulated by ξ :

$$g_{\nu,\xi}(t) := e^{2i\pi\xi t} g(t - \nu).$$

The resulting short–time Fourier transform of $f \in L_2(\mathbb{R})$ is

$$\tilde{S}f(\nu, \xi) := \langle f, g_{\nu,\xi} \rangle = \int_{\mathbb{R}} f(t) g(t - \nu) e^{-2i\pi\xi t} dt.$$

The notion of *windowed* or *short–time Fourier transform* (STFT) originates from the fact that multiplication by $g(t - \nu)$ localizes the Fourier transform in the neighborhood of $t = \nu$. The STFT thus allows the localization of a frequency phenomenon within a certain time window, a property that is nonexistent for the Fourier transform. This is why comparisons between the wavelet transform and the Fourier transform are usually restricted to the special case of the STFT.

The calculation of the Heisenberg boxes of time–frequency uncertainty for the STFT reveals the following: The window g is even, thus $g_{\nu,\xi}$ is centered at ν and the time–spread around ν is independent of ν and ξ :

$$\sigma_t^2 = \int_{\mathbb{R}} (t - \nu)^2 |g_{\nu,\xi}(t)|^2 dt = \int_{\mathbb{R}} t^2 |g(t)|^2 dt.$$

The Fourier transform of $g_{\nu,\xi}$ is

$$\hat{g}_{\nu,\xi}(\omega) = e^{-i\nu(\omega - \xi)} \hat{g}(\omega - \xi),$$

and its center frequency is ξ . The frequency spread around ξ is

$$\sigma_\omega^2 = \frac{1}{2\pi} \int_{\mathbb{R}} (\omega - \xi)^2 |\hat{g}_{\nu,\xi}(\omega)|^2 d\omega = \frac{1}{2\pi} \int_{\mathbb{R}} \omega^2 |\hat{g}(\omega)|^2 d\omega$$

and is independent of ν and ξ . Consequently, the Heisenberg box of the translated and modulated window $g_{\nu,\xi}$ has the area $\sigma_t \sigma_\omega$, centered at (ν, ξ) . The size of this box is independent of ν and ξ . This means a short-time Fourier transform has identical resolution across the time–frequency plane (see Figure 1.3 (a)). [Mal98]

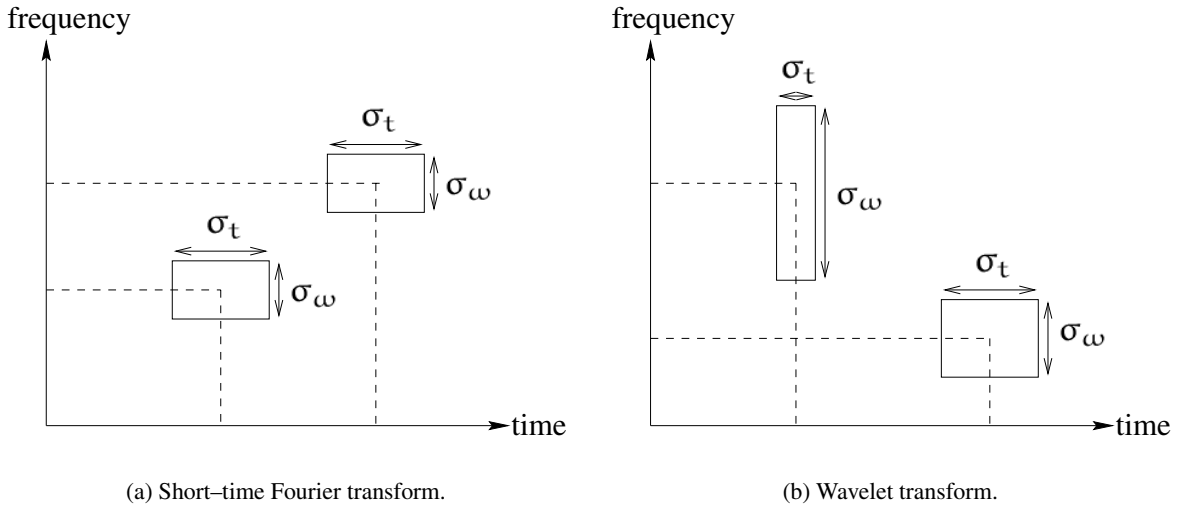


Figure 1.3: Time–frequency resolution, visualized by Heisenberg boxes of uncertainty. (a) Short-time Fourier transform: The shape of the boxes depends uniquely on the choice of window. In higher frequencies, the analyzing function oscillates stronger, covering the same time. (b) Wavelet transform: the area of uncertainty remains constant, but its time, respectively, frequency resolution varies.

1.4.3 Properties of the Wavelet Transform

The time–frequency resolution of the wavelet transform $\tilde{f}_\psi(a, b)$ depends on the time–frequency spread of the wavelet atoms $\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$ (see Equation (1.4)). In the following, we want to express the resolution of the dilated and translated wavelets $\psi_{a,b}$ in terms of the mother wavelet ψ . For simplicity, we assume that ψ is centered at 0, thus $\psi_{a,b}$ is centered at $t = b$. A change of variables leads to

$$\int_{\mathbb{R}} (t - b)^2 |\psi_{a,b}(t)|^2 dt = a^2 \int_{\mathbb{R}} t^2 |\psi(t)|^2 dt = a^2 \sigma_t^2.$$

The Fourier transform of $\psi_{a,b}$ is a weighted dilation of $\hat{\psi}$ by $1/a$: $\hat{\psi}_{a,b}(\omega) = \sqrt{a} e^{-2i\pi\omega b} \hat{\psi}(a\omega)$. Its center frequency is therefore η/a , where η denotes the center frequency of $\hat{\psi}$. The energy spread of

$\hat{\psi}_{a,b}$ around η/a is

$$\frac{1}{2\pi} \int_0^\infty \left(\omega - \frac{\eta}{a}\right)^2 |\hat{\psi}_{a,b}(\omega)|^2 d\omega = \frac{1}{a^2} \frac{1}{2\pi} \int_0^\infty (\omega - \eta)^2 |\hat{\psi}(\omega)|^2 d\omega = \frac{\sigma_\omega^2}{a^2}.$$

The energy spread of a wavelet atom $\psi_{a,b}$ is thus centered at $(a, \frac{\eta}{a})$ and of size $a\sigma_t$ along time and σ_ω/a along frequency. [Mal98]

The area of the Heisenberg box of uncertainty remains equal to

$$(a\sigma_t) \times (\sigma_\omega/a) = \sigma_t\sigma_\omega$$

at all scales. It is especially independent of the translation parameter b . However, the resolution in time and frequency depends on the scaling parameter a (see Figure 1.3 (b)).

1.5 Sampling Grid of the Wavelet Transform

In this section we discuss the question of invertibility of the wavelet transform with minimal redundancy. Questions such as *what conditions have to be satisfied to be able to reconstruct the original signal f from its wavelet transform* and *does \tilde{f}_ψ have to be known for each pair of dilation and translation parameters $(a, b) \in \mathbb{R}_{>0} \times \mathbb{R}$ to be able to reconstruct f* motivate this section.

In Equation (1.5) we have seen that the set of *dilated* and *translated* versions of a single wavelet presents a basis in $L_2(\mathbb{R})$. Mallat [Mal98] has shown that the parameter a , which steers the dilation of the wavelet ψ (see Equation (1.4)), can be restricted further. In effect, the *dyadic* wavelet transform of $f \in L_2(\mathbb{R})$,

$$\tilde{f}_\psi(2^j, b) = \frac{1}{\sqrt{2^j}} \int_{\mathbb{R}} f(t) \psi^* \left(\frac{t-b}{2^j} \right) dt, \quad (1.6)$$

defines a complete and stable representation of f if the frequency axis is completely covered. This means it is sufficient to look at the wavelet transform at the dilation steps 2^j .

If the Fourier transform of a wavelet ψ has finite support, then this band-limited function can be sampled without loss of information. Shannon's sampling theorem gives the critical sampling rate for ψ which is necessary to allow perfect reconstruction [Ste00]. Every dilated version of a band-limited wavelet is again band-limited, and therefore it may be sampled without loss of information with a sampling frequency of 2^j :

Theorem 1.1 *In order to reconstruct a wavelet-analyzed function f , it is sufficient to know the values of the wavelet transform \tilde{f}_ψ on a grid [Hol95]*

$$\left\{ (2^j, n2^j) : j, n \in \mathbb{Z} \right\}.$$

Theorem 1.1 says that even the translation parameter b in the definition of the dyadic wavelet transform (1.6) can be restricted further, and the sampling distance of the translation parameter depends on the underlying scale. This means that we obtain the critically sampled wavelet transform as a grid in the (a, b) -plane with the dilation parameter set to $a = 2^j$ and the translation parameter set to $b = n2^j$. This *dyadic grid* is given in Figure 1.4 (b).

In comparison to the sampling grid of the dyadic wavelet transform, Figure 1.4 (a) demonstrates a sampling grid for the short-time Fourier transform. The sampling grids correspond to the time–frequency spread of the short-time Fourier and the wavelet transforms in Figure 1.3 when the sampling points are interpreted as center points of their respective Heisenberg boxes.

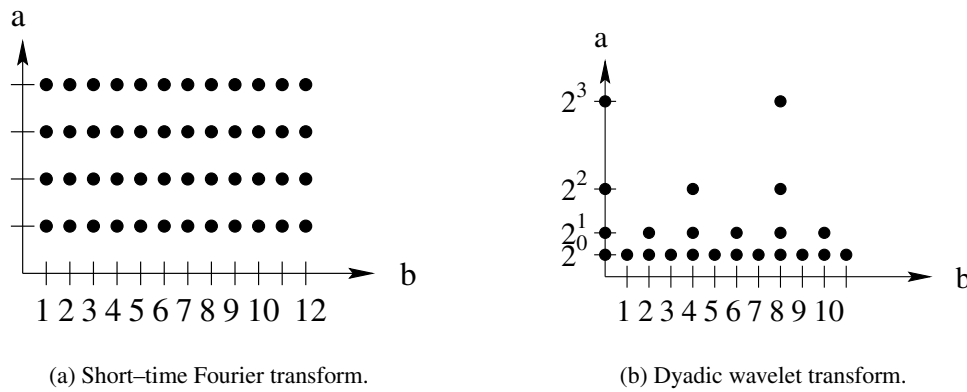


Figure 1.4: Sampling grids of the short-time Fourier and the dyadic wavelet transforms. The sampling intervals in (a) remain constant all over the (a, b) -plane, while the grid in (b) depends on the scale.

The fact that the translation parameter b doubles from one iteration step to the next can now successfully be exploited with the idea of *downsampling* a discrete signal.

Example 1.1 *Let us consider a discrete signal with 16 coefficients at level 2^j . The signal is wavelet-transformed at level 2^j . At the following level 2^{j+1} , only every second sample of the first iteration will be used (see Figure 1.4 (b)). In practice this means that the signal will be downsampled by factor 2 and only 8 coefficients will enter the second iteration. After four iterations, the original signal will be represented by only one remaining coefficient.*

Obviously, the iteration of the wavelet transform described in Example 1.1 loses some information since a signal of one single coefficient contains less detailed information than the original signal of 16 coefficients. The inferred question of how to preserve the detailed information is addressed in the following section on multiscale analysis, which leads to the fast wavelet transform.

1.6 Multiscale Analysis

The *multiscale analysis* (MSA) of a signal f is based on successive decomposition into a series of *approximations* and *details* which become increasingly coarse. At the beginning, the signal is split

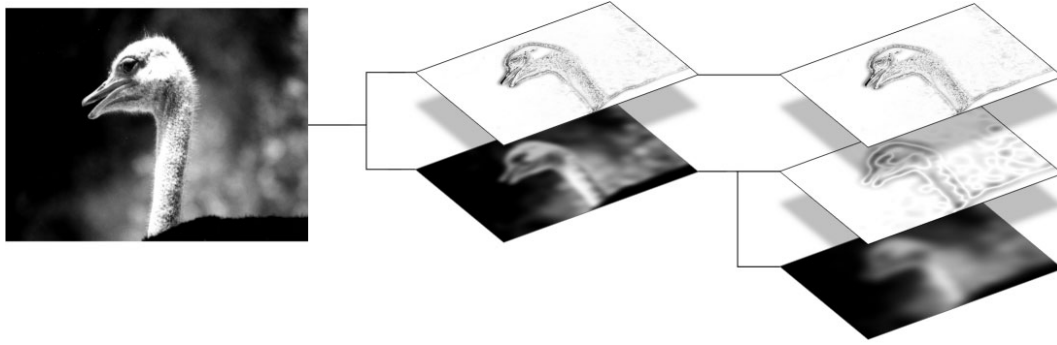


Figure 1.5: Multiscale analysis. The image is subdivided into approximations and details. While the approximation contains a coarser version of the original, the details contain all the information that has been lost in the approximation. Iteration starts on the approximations. The high-pass filtered parts in each iteration are band-pass filtered parts when seen in the overall process.

into an approximation and a detail that together yield the original. The subdivision is such that the approximation signal contains the low frequencies, while the detail signal collects the remaining high frequencies. By repeated application of this subdivision rule on the approximation, details of increasingly coarse resolution are separated out, while the approximation itself grows coarser and coarser. Figure 1.5 demonstrates the idea of this algorithm. The original image *Ostrich* is presented on the left. In the middle of Figure 1.5, the first decomposition step is visualized, where a coarse resolution of the *Ostrich* plus the fine-grained details together form the original. On the right hand side, this same procedure has been repeated on the approximation of the first decomposition. The detail that had been separated out in the first step is kept unchanged, while the approximation of the first step is treated as the heretofore original, i.e., a detail image at this level is separated out, and an even coarser approximation remains. The original *Ostrich* is obtained by ‘looking at all three versions of the image from bottom to top and adding up the approximation plus the two detail images’.

Multiscale analysis was linked to the wavelet theory by Mallat [Mal87] [Mal89] [Mal98] by introducing a new function, the *scaling function* φ . The MSA enables the construction of fast algorithms for the analysis and synthesis with wavelets and even the definition of wavelet bases. Finally, it allows a review of the fast wavelet transform of Section 1.5 and a recovery of the missing detail information of Example 1.1.

In multiscale analysis, a signal $f \in L_2(\mathbb{R})$ is projected onto a subspace of $L_2(\mathbb{R})$. Since in the *dyadic* approach mentioned earlier, the resolution from one iteration to the next one is varied by the factor 2 (see Theorem 1.1), we restrict our considerations to subspaces $V_{2^j} \subset L_2(\mathbb{R})$. The projection separates out the details of the signal and keeps only the approximation on level 2^j . Renewed application of this procedure on the approximation gives a new subspace $V_{2^{j+1}}$:

$$V_{2^j} = V_{2^{j+1}} \oplus W_{2^{j+1}}, \quad j = 0, 1, \dots \quad (1.7)$$

We consider the projection operator P_{2^j} , which maps the signal f onto V_{2^j} . Via successive projection of f onto subspaces of $L_2(\mathbb{R})$, we obtain a series of resolutions $\{2^j\}_{j \in \mathbb{Z}}$, where the resolution decreases with increasing 2^j .

In order for the multiscale approach to approximate a given function $f \in L_2(\mathbb{R})$ with arbitrary precision, four conditions are sufficient that influence the relationship of the subspaces among themselves:

1. The scaling function is orthogonal to its translates by integers. In other words, the inner products of φ and its integer translates vanish.
2. An approximation at the given resolution 2^j contains all information necessary to specify the next coarser approximation 2^{j+1} .
3. The multiscale analysis is a series of closed nested subspaces:

$$\{0\} \subset \dots \subset V_{2^{j+1}} \subset V_{2^j} \subset V_{2^{j-1}} \subset \dots \subset L_2(\mathbb{R}).$$

Furthermore, the intersection of all subspaces contains only the function 0, and their union is *dense* in $L_2(\mathbb{R})$, i.e., all functions in all subspaces can approximate every function in $L_2(\mathbb{R})$:

$$\begin{aligned} \bigcap_{j \in \mathbb{Z}} V_{2^j} &= \{0\}, \\ \overline{\bigcup_{j \in \mathbb{Z}} V_{2^j}} &= L_2(\mathbb{R}). \end{aligned}$$

4. The approximation at a given resolution is self-similar to other resolutions:

$$f(\cdot) \in V_{2^0} \Leftrightarrow f(2^{-j}\cdot) \in V_{2^j}. \quad (1.8)$$

If these relationships between the spaces $\{V_{2^j}\}_{j \in \mathbb{Z}}$ are met, we are dealing with a *multiscale approximation* of $L_2(\mathbb{R})$.

1.6.1 Approximation

Our goal is to approximate an arbitrary signal $f \in L_2(\mathbb{R})$ by coarser signals within each subspace V_{2^j} of $L_2(\mathbb{R})$. Therefore, we need basis functions of these subspaces. That is, we are looking for functions that span the subspaces in each resolution (or scale). The projection of f onto a subspace V_{2^j} would then possess an explicit notation. The following theorem shows that there exists an orthonormal basis of V_{2^j} which is defined through dilation and translation of a single function:

Theorem 1.2 *Let $\{V_{2^j}\}_{j \in \mathbb{Z}}$ be a multiscale approximation in $L_2(\mathbb{R})$. Then there exists a single function $\varphi \in L_2(\mathbb{R})$ such that*

$$\left\{ \frac{1}{\sqrt{2^j}} \varphi \left(\frac{t - k2^j}{2^j} \right) \right\}_{j,k \in \mathbb{Z}}$$

is an orthonormal basis of V_{2^j} .

φ is called the *scaling function*. It is the counterpart to the wavelets which we will define later in this section. The explicit form of the scaling function in V_{2^0} , for example, results from $V_{2^0} \subset V_{2^{-1}}$ and Theorem 1.2 and is written as a recursive difference:

$$\varphi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_0[k] \varphi(2t - k). \quad (1.9)$$

Equation (1.9) is called the *scaling equation* as it connects the basis function at a given resolution to the basis function at a resolution twice as high. h_0 is called the *filter mask* for the scaling function φ (see also the definition of h_0 within the general filter theory in Section 2.2.1). Its discrete filter coefficients depend on the choice of the function φ .

Two examples illustrate the construction of the filter mask, when a function at a given resolution is represented by means of its various translates at double that resolution.

Example 1.2 Let φ be the indicator function on the half-open interval $[0, 1[$, i.e., $\varphi = \mathbf{1}_{[0,1[} \in V_{2^0}$. On a double fine scale, $\varphi(t)$ would need two representatives, i.e.,

$$\varphi(t) = \varphi(2t) + \varphi(2t - 1).$$

Here, the filter coefficients are: $h_0[0] = h_0[1] = 1$ and $h_0[k] = 0$ for $k \neq 0, 1$. See also Figure 1.6 (a).

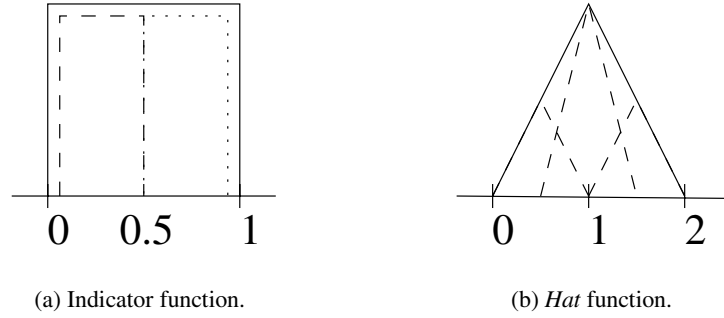


Figure 1.6: Scaling equation. In (a), the original indicator function is the sum of two finer indicator functions (here: dashed and dotted). For better visualization, the functions in the fine scales are not painted directly over the original signal. In (b), the *hat* that goes from 0 to 2 is the original function. The three dashed *hats* are translated versions in the double fine scale, where the left and the right *hats* are scaled by the factor $\frac{1}{2}$.

Example 1.3 Let φ be the hat function

$$\varphi(t) = \left\{ \begin{array}{lll} t & : & t \in [0, 1[, \\ 2 - t & : & t \in [1, 2[, \\ 0 & : & \text{else.} \end{array} \right\} \in V_{2^0}.$$

On the double fine scale, $\varphi(t)$ would need three representatives, i.e.,

$$\varphi(t) = \frac{1}{2}\varphi(2t) + \varphi(2t-1) + \frac{1}{2}\varphi(2t-2).$$

Here, the filter coefficients are: $h_0[0] = h_0[2] = \frac{1}{2}$, $h_0[1] = 1$ and $h_0[k] = 0$ else. See also Figure 1.6 (b).

Apart from these two rather intuitive examples, the filter coefficients h_0 in Equation (1.9) are calculated from the inner product of the scaling function and its dilated and translated versions:

$$h_0[k] = \langle \varphi(t), \sqrt{2}\varphi(2t-k) \rangle.$$

The conditions for the scaling function and their implications on the filter coefficients are [Boc98]:

$$\int_{\mathbb{R}} \varphi(t) dt = 1 \rightarrow \sum_{k \in \mathbb{Z}} h_0[k] = \sqrt{2}, \quad (1.10)$$

$$\int_{\mathbb{R}} \varphi(t) \varphi^*(t-y) dt = \delta_{0y} \rightarrow \sum_{k \in \mathbb{Z}} h_0[k] h_0^*[k-2y] = \delta_{0y}, \quad (1.11)$$

$$\{\varphi(t-y)\} \text{ orthonormal basis of } V_{2^0} \rightarrow \sum_{k \in \mathbb{Z}} |h_0[k]|^2 = 1, \quad (1.12)$$

where $y \in \mathbb{Z}$.

We are now interested in an explicit form of the approximation of a signal f in a space V_{2^j} . This approximation is the projection $P_{2^j} f$ of f . It results from the decomposition of f with regard to the orthonormal basis of Theorem 1.2:

$$\begin{aligned} (P_{2^j} f)(t) &= \sum_{k \in \mathbb{Z}} \left\langle f(t), \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t-k2^j}{2^j}\right) \right\rangle \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t-k2^j}{2^j}\right) \\ &=: \sum_{k \in \mathbb{Z}} \mathcal{A}_{f,k}^j \frac{1}{\sqrt{2^j}} \varphi\left(\frac{t-k2^j}{2^j}\right). \end{aligned} \quad (1.13)$$

The series $\{\mathcal{A}_{f,k}^j\}_{k \in \mathbb{Z}}$ provides a discrete approximation of f at the scale 2^j . This approximation is uniquely determined by the calculation of the inner product (1.13), where the scaling function enters.

1.6.2 Detail

Until now, we have discussed the scaling function φ , its filter mask h_0 , and the approximation of a signal in a given subspace. With these approximations, we lose information about the original signal. As mentioned above, the difference between succeeding approximations of the resolutions 2^{j-1} and

2^j is referred to as *detail* information of level 2^j . These details are exactly the information that is lost during approximation. If this detail space is denoted by W_{2^j} , the finer space $V_{2^{j-1}}$ is a direct sum (see also Equation (1.7)):

$$V_{2^{j-1}} = V_{2^j} \oplus W_{2^j}. \quad (1.14)$$

Since Equation (1.14) holds on every level, V_{2^j} and thus $V_{2^{j-1}}$ can be further subdivided:

$$\begin{aligned} V_{2^{j-1}} &= V_{2^j} \oplus W_{2^j} \\ &= V_{2^{j+1}} \oplus W_{2^{j+1}} \oplus W_{2^j} \\ &= \dots \\ &= V_{2^J} \oplus W_{2^J} \oplus \dots \oplus W_{2^{j+1}} \oplus W_{2^j}, \end{aligned} \quad (1.15)$$

where 2^J is an arbitrary stopping index. Obviously,

$$V_{2^J} = \bigoplus_{j \geq J+1} W_{2^j}$$

is the ‘collection’ of all the decompositions that are not explicitly carried out, and the complete space could also be represented in terms of details only:

$$L_2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_{2^j}.$$

The spaces W_{2^j} inherit the self-similarity of the spaces V_{2^j} in Equation (1.8),

$$f(\cdot) \in W_{2^0} \Leftrightarrow f(2^{-j}\cdot) \in W_{2^j}.$$

Analogous to Theorem 1.2, an orthonormal basis for the detail spaces exists:

Theorem 1.3 *Let $\{W_{2^j}\}_{j \in \mathbb{Z}}$ be a multiscale analysis in $L_2(\mathbb{R})$. Then a single function $\psi \in L_2(\mathbb{R})$ exists such that*

$$\left\{ \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - k2^j}{2^j} \right) \right\}_{j,k \in \mathbb{Z}} \quad (1.16)$$

is an orthonormal basis of W_{2^j} .

ψ is called an *orthogonal wavelet*. Analogous to the scaling Equation (1.9), again a recursive difference equation on the wavelets exists, the *wavelet equation*:

$$\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_1[k] \varphi(2t - k),$$

where the coefficients of the filter mask h_1 for the wavelet ψ are calculated as

$$h_1[k] = \langle \psi(t), \sqrt{2}\varphi(2t - k) \rangle.$$

As we have stated conditions for the scaling function and its filter mask in Equations (1.10)–(1.12), the following conditions hold for the wavelet and its filter mask:

$$\int_{\mathbb{R}} \psi(t) dt = 0 \rightarrow \sum_{k \in \mathbb{Z}} h_1[k] = 0, \quad (1.17)$$

$$\int_{\mathbb{R}} \psi(t) \psi^*(t - y) dt = \delta_{0y} \rightarrow \sum_{k \in \mathbb{Z}} h_1[k] h_1^*[k - 2y] = \delta_{0y}, \quad (1.18)$$

$$\{\psi(t - y)\} \text{ orthonormal basis of } W_{2^0} \rightarrow \sum_{k \in \mathbb{Z}} |h_1[k]|^2 = 1. \quad (1.19)$$

Like in Equation (1.13), we are interested in an explicit form to describe the detail of a signal f in the detail space W_{2^j} . This projection results from the decomposition of f with regard to the orthonormal basis in Theorem 1.3:

$$\begin{aligned} (P_{2^{j-1}} f)(t) - (P_{2^j} f)(t) &= \sum_{k \in \mathbb{Z}} \left\langle f(t), \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - k2^j}{2^j} \right) \right\rangle \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - k2^j}{2^j} \right) \\ &=: \sum_{k \in \mathbb{Z}} \mathcal{D}_{f,k}^j \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - k2^j}{2^j} \right). \end{aligned}$$

1.6.3 Summary and Interpretation

After detailing the underlying concept of approximating a signal on different scales, this section gives a concluding overview. Table 1.1 summarizes the relation between signal and spaces.

Signal	Space
Original signal $f(t)$	$L_2(\mathbb{R})$
Approximation at Level 2^j	V_{2^j}
Detail at level 2^j	W_{2^j}
Relation between the approximation levels	$V_{2^j} = V_{2^{j+1}} \oplus W_{2^{j+1}}$
Signal is the sum of all its details	$L_2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_{2^j}$
Decomposition of the signal	$L_2(\mathbb{R}) = V_{2^J} \oplus \bigoplus_{j < J} W_{2^j}$

Table 1.1: Relations between signals and spaces in multiscale analysis.

What we have already seen in Equation (1.15), and what is again written in Table 1.1, is that the original signal could be decomposed into an infinite sum of details. In practical considerations, only finite sums will be calculated though, and this is where the scaling function comes in. The scaling

function defines the approximation at the ‘stopping level’. It thus defines the resolution of the coarsest approximation. Without this, the starting points of all the above considerations would be undefined. In the context of the *Ostrich* in Figure 1.5, the scaling function determines the coarse approximation on the bottom right, while the wavelets determine the details on the two levels on the top and middle right.

Since we concentrate on resolutions that are bisected in each iteration, Figure 1.7 gives another heuristic for the multiscale algorithms, regarded as low-pass filters, i.e., scaling functions, and high-pass filters, respectively, band-pass filters, i.e., wavelets in the frequency domain.

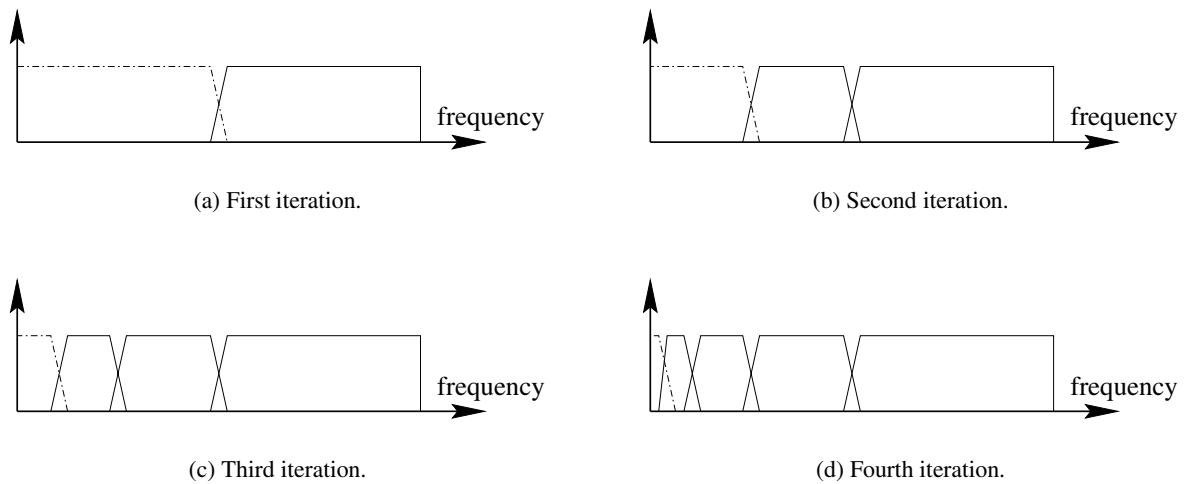


Figure 1.7: Subband coding. In each iteration, half the resolution is ‘separated out’ as details. The remaining approximation is then further subdivided. In each iteration, the scaling function determines the remaining approximation that subsummarizes all the yet unconsidered parts (here marked in dashed lines).

If we now recall our considerations of the time–frequency resolution of the wavelet Heisenberg boxes in Section 1.4 and of the sampling grid of the dyadic wavelet transform in Section 1.5, we discover the following. The multiscale analysis bisects the frequency resolution of a given signal f at every iteration step. On the other hand, the overall time–frequency resolution of a wavelet Heisenberg box has to remain constant. Consequently, when we switch from resolution V_{2^j} to the next coarser resolution $V_{2^{j+1}}$, the time–spread is doubled. Conversely, switching from resolution V_{2^j} to the next finer resolution $V_{2^{j-1}}$ cuts in half the ‘region of uncertainty’. If the complete frequency spectrum of a signal is set in relation to its complete time coverage, and the time–frequency Heisenberg boxes of this dyadic wavelet transform are painted, a tiling of the time–frequency plane like in Figure 1.8 results. The role of the scaling function in Figure 1.8 is to determine the very lowest box (painted dotted).

Another phenomenon finds explanation in Figure 1.8. Since the information of a wavelet–transformed signal is given as coefficients in the time–frequency domain, the *time influence* of such a coefficient broadens the lower its frequency is. In other words, a coefficient in the upper row of Figure 1.8 influences only half as many coefficients of the original signal as does a coefficient in the second row from the top. The two lower rows in Figure 1.8 contain coefficients whose influence on the original signal is four times that of those in the upper row.

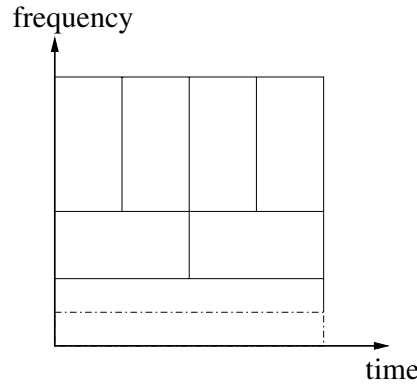


Figure 1.8: Tiling the time–scale domain for the dyadic wavelet transform. The iteration has been carried out three times, thus bisecting the overall frequency spread three times. Each division of frequency spread, however, results in a doubling of the time–spread.

1.6.4 Fast Wavelet Transform

The multiscale analysis presented above provides a simple and rapid method of decomposing a signal f into its components of different resolutions: The approximation on each scale ‘relieves’ the signal of its details. This algorithm successively separates out the details, beginning with very fine details in the first iteration, and succeeding with coarser and coarser details. At each step, the detail information is encoded with the help of the wavelet function ψ , i.e., with the filter mask h_1 induced by ψ (see Section 1.6.2). The scaling function φ is used to encode the approximation. Thus, translated versions of the scaling function at a given scale approximate the signal at the given resolution, while dilated versions make an image of the signal at resolution twice as coarse. In signal processing, this is accomplished by application of the filter mask h_0 induced by φ (see Section 1.6.1). The next step of half the resolution is executed on the ‘decimated’ signal by taking one sample out of two. This decimation is referred to as *subsampling* a signal.

Since each iteration halves the number of signal samples, the signal is quickly reduced to the very coarsest approximation of one coefficient only, denoting the average value. As all discarded information is encoded in the details, the process is losslessly reversible. This *fast wavelet transform* can be interpreted as building averages of neighboring coefficients (thus, additions), and calculating the differences of the coefficients towards these averages (thus, subtractions). The averages form the approximations, while the differences form the details. This is the meaning of the formula

$$V_{2^{-1}} = V_{2^J} \oplus W_{2^J} \oplus \dots \oplus W_{2^1} \oplus W_{2^0}, \quad (1.20)$$

which we have already seen in Equation (1.15).

1.7 Transformation Based on the Haar Wavelet

After the discussion of the theory of wavelets in the previous sections, this section exemplifies the fast wavelet transform with the Haar wavelet. The Haar transform is appropriate to introduce the

philosophy and nature of a wavelet transform, as it contains an intrinsically intuitive interpretation. No previous knowledge of wavelet theory is necessary to understand this section. Only at the end of this section will we bridge the gap between this example and the general theory in Section 1.6. As a brief outlook, the filters encountered in this section will be put into the general context of orthogonal wavelet filters, which are discussed in Chapter 2.

Suppose we have a one-dimensional discrete signal f (e.g., an audio signal) with sampling distance 1. The aim is to decompose f into coefficients in the time-scale domain by means of the wavelet transform. Therefore, we are interested in an algorithm that can ‘decorrelate’ the signal. In other words, we would like to express the same information with a signal of fewer coefficients, i.e., coefficients with a larger sampling distance. While we accept that the coarser signal will not *exactly* represent the signal, it should represent at least an *approximation* with an acceptable error, i.e., the difference between the approximating signal and the original ought to be small. Finally, we demand that we will be able to trace the error. Thus, we want to have precise control over the information that is lost by using fewer coefficients.

A first attempt — and the most intuitive one — is to reduce the number of samples by the factor 2 and to calculate the mean value, which we will call *approximation*, of each two neighboring samples (see Figure 1.9). Expressed in terms of filters, the calculation of the approximation is carried out with $[\frac{1}{2}, \frac{1}{2}]$. The information that has been lost is the difference between the original signal value and this average. Note that the difference between the first value of each pair and the approximation is the negative of the difference between it and the second value of each pair. It is thus sufficient to store *one* of the two differences, which we will call *detail*. These details are given in Figures 1.9 (a) and (c). The filter for the detail calculation is given as $[\frac{1}{2}, -\frac{1}{2}]$.

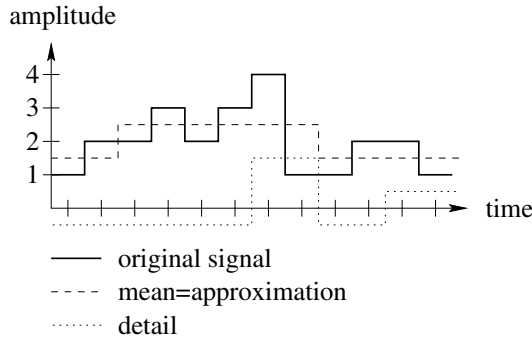
Figure 1.9 (c) also demonstrates that the total number of coefficients needed to describe the original signal has not changed. The original 12 samples in our example have been replaced by 6 approximation coefficients and 6 detail coefficients. Only the ‘coordinate system’ has changed, i.e., we have changed the basis under consideration.

The process of approximating the original signal and storing the details apart is now iterated over the approximations. Figures 1.9 (b) and (d) show the first two, respectively, three, iteration steps on our sample signal. As we have shown in Section 1.6.3, the influence of a single coefficient of the transformed space on the original signal broadens with increasing iteration depth. Here, we recover the notion of *multiresolution* from Section 1.6.

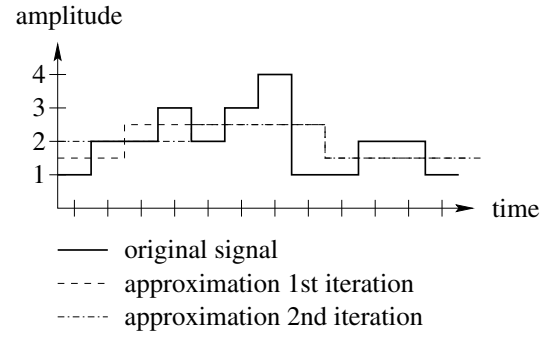
The steps performed until now are part of the decomposition or *analysis*. The *synthesis*, i.e., reconstruction of the original signal from its coefficients in the transformed space, however, is easy. We detail the calculation on the first iteration of our sample signal in Figure 1.9. To recover the first two samples of our signal, we consider the first approximation and detail values in Figure 1.9 (c), i.e., 1.5 as approximation and -0.5 as detail:

$$\begin{aligned} 1.5 \cdot 1 + (-0.5) \cdot 1 &= 1 && \text{synthesis of } 1^{st} \text{ value} \\ 1.5 \cdot 1 - (-0.5) \cdot 1 &= 2 && \text{synthesis of } 2^{nd} \text{ value} \end{aligned}$$

Analogously, the syntheses of the third and fourth signal entries are recovered from the second approximation (i.e., 2.5) and the second detail (i.e., -0.5). The corresponding filters are $[1, 1]$ for the synthesis of the first value of each pair, and $[1, -1]$ for the synthesis of the second value of each pair.



(a) Graphical illustration, level 1.



(b) Graphical illustration, level 2.

1	2	2	3	2	3	4	1	1	2	2	1
1.5	2.5	2.5	2.5	1.5	1.5						
-0.5	-0.5	-0.5	1.5	-0.5	-0.5						

original signal
mean=approximation
detail

(c) Coefficients, level 1.

1	2	2	3	2	3	4	1	1	2	2	1
1.5	2.5	2.5	2.5	1.5	1.5						
2			2.5			1.5					
			2.25								

original signal
approximation 1st iteration
approximation 2nd iteration
approximation 3rd iteration

(d) Coefficients, levels 2 and 3.

Figure 1.9: Haar transform of a one-dimensional discrete signal. (a) and (c): start of the iteration; including details. (b) and (d): levels 2 and 3; only the approximations are shown.

In the total, we have performed the Haar transform on a one-dimensional discrete signal. The interpretation as approximation and detail is intuitive. We have seen that the total number of coefficients remains unchanged by the transform, which explains why we call it a basis transform. Furthermore, we have seen that the analysis requires two filters, an approximation (i.e., low-pass) filter and a detail (i.e., high-pass) filter. The synthesis also requires two filters, one for the synthesis of the even samples (i.e., inverse low-pass filter), and one for the synthesis of the odd samples (i.e., inverse high-pass filter). Finally, we have seen that procedure is lossless.

The linking of the Haar transform to filter masks constructed by multiscale analysis, and characterized in Section 1.6 through Equations (1.10)–(1.12) and (1.17)–(1.19) is done quickly. The above Haar analysis and synthesis filters already accomplish the condition (1.17) for the coefficients of the high-pass filter:

- $\sum_{k \in \mathbb{Z}} h_1[k] = \frac{1}{2} - \frac{1}{2} = 0.$

We are still missing other properties, two of which for the low-pass, and one for the high-pass filter coefficients:

- $\sum_{k \in \mathbb{Z}} h_0[k] = \sqrt{2},$
- $\sum_{k \in \mathbb{Z}} |h_0[k]|^2 = 1,$ and
- $\sum_{k \in \mathbb{Z}} |h_1[k]|^2 = 1.$

By simply shifting the factor $\frac{1}{\sqrt{2}}$ from the analysis filters to the synthesis filters, the filters used in the above sample Haar transform become

$$\begin{array}{ll} \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] & \text{low-pass filter,} \\ \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] & \text{high-pass filter,} \\ \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] & \text{inverse low-pass filter, and} \\ \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] & \text{inverse high-pass filter.} \end{array}$$

In this form, direct interpretation of the filter coefficients has vanished. However, the filter coefficients fit perfectly into the general theory presented in the following chapter. In the literature, the Haar wavelet filter can be found in both writings.

Chapter 2

Filter Banks

And since geometry is the right foundation of all painting, I have decided to teach its rudiments and principles to all youngsters eager for art.

– Albrecht Dürer

2.1 Introduction

An understanding of filter banks is crucial for a profound understanding of low-pass and high-pass filters and their design. In this chapter we elaborate the conditions that the filter masks h_0 and h_1 of the multiscale analysis have to satisfy.

”The purpose of subband filtering is of course not to just decompose and reconstruct. The goal of the game is to do some compression or processing between the decomposition and reconstruction stages. For many applications, compression after subband filtering is more feasible than without filtering. Reconstruction after such compression schemes (quantization) is then not perfect any more, but it is hoped that with specially designed filters, the distortion due to quantization can be kept small, although significant compression ratios are obtained.”

— I. Daubechies [Dau92]

The theory of filter banks requires the notion of filters. We start with the consideration of a *band-limited* function $g \in L_2(\mathbb{R})$, i.e., its Fourier transform has compact support $[-L, L]$. Shannon’s sampling theorem [Dau92] [Ste00] then indicates for every $M \geq L$

$$g(x) = \sum_{k \in \mathbb{Z}} g\left(\frac{k}{2M}\right) \text{sinc}(2Mx - k), \quad (2.1)$$

where the sinc function is defined as $\text{sinc}(x) = \frac{\sin(x)}{x}$. For the sake of simplicity, we set $L = \frac{1}{2}$. Then for $\omega \in [-\frac{1}{2}, \frac{1}{2}]$, the Fourier transform of g can be represented by its Fourier series

$$\hat{g}(\omega) = \sum_{k \in \mathbb{Z}} g[k] e^{-2\pi i k \omega},$$

where

$$g[k] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{g}(\omega) e^{2\pi i k \omega} d\omega = \int_{-\infty}^{\infty} \hat{g}(\omega) e^{2\pi i k \omega} d\omega = g(k).$$

2.2 Ideal Filters

2.2.1 Ideal Low-pass Filter

A low-pass filter leaves all low frequencies of a signal unchanged, while ‘cutting off’ high frequencies. The Fourier transform \hat{h}_0 of an *ideal* low-pass filter h_0 has no transition between the cut-off frequency and the pass-frequency, therefore (see Figure 2.1 (a))

$$\hat{h}_0(\omega) = \begin{cases} 1 & : \omega \in [-\frac{1}{4}, \frac{1}{4}] \\ 0 & : \omega \in [-\frac{1}{2}, -\frac{1}{4}[\text{ or }]\frac{1}{4}, \frac{1}{2}]. \end{cases}$$

We are now interested in the (discrete) *filter mask* $h_0[\cdot]$ corresponding to the filter \hat{h}_0 , i.e., in the Fourier coefficients required:

$$\begin{aligned} h_0[k] &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{h}_0(\omega) e^{2\pi i k \omega} d\omega = \int_{-\frac{1}{4}}^{\frac{1}{4}} e^{2\pi i k \omega} d\omega \\ &\stackrel{k \neq 0}{=} \frac{1}{2\pi i k} e^{2\pi i k \omega} \Big|_{-\frac{1}{4}}^{\frac{1}{4}} = \frac{\sin(\pi k/2)}{\pi k}. \end{aligned}$$

The coefficients of the filter mask are thus:

$$h_0[0] = \frac{1}{2}, \quad h_0[2k] = 0, \quad h_0[2k+1] = \frac{(-1)^k}{(2k+1)\pi}, \quad (2.2)$$

and the low-pass filter can be written as

$$\hat{h}_0(\omega) = \sum_{m \in \mathbb{Z}} h_0[m] e^{-2\pi i m \omega} = \frac{1}{2} + \sum_{k \in \mathbb{Z}} \frac{(-1)^k}{(2k+1)\pi} e^{-2\pi i (2k+1)\omega}. \quad (2.3)$$

The application of an ideal low-pass filter to a function f means multiplication of $\hat{h}_0(\omega)$ and $\hat{f}(\omega)$ in the frequency space, which corresponds to convolution in the time domain with subsequent transformation into the frequency domain:

$$\begin{aligned}
 \widehat{h_0 * f}(\omega) &= \hat{h}_0(\omega) \hat{f}(\omega) \\
 &= \sum_{m \in \mathbb{Z}} h_0[m] e^{-2\pi i m \omega} \sum_{n \in \mathbb{Z}} f[n] e^{-2\pi i n \omega} \\
 &= \sum_{k \in \mathbb{Z}} \left(\sum_{j \in \mathbb{Z}} h_0[k - j] f[j] \right) e^{-2\pi i k \omega} \\
 &= \sum_{k \in \mathbb{Z}} f_{\text{low}}[k] e^{-2\pi i k \omega} \\
 &= \hat{f}_{\text{low}}(\omega).
 \end{aligned}$$

This is the Fourier series of the low-pass filtered function f_{low} , where

$$f_{\text{low}}[k] = \sum_{j \in \mathbb{Z}} h_0[k - j] f[j]. \quad (2.4)$$

Due to the application of the low-pass filter, we have support $\hat{f}_{\text{low}} \subseteq [-\frac{1}{4}, \frac{1}{4}]$. The Shannon sampling theorem (Equation (2.1)) for $M = L = \frac{1}{4}$ says that the bandwidth-reduced f_{low} can be reconstructed from the subsampled one by

$$f_{\text{low}}(t) = \sum_{k \in \mathbb{Z}} f_{\text{low}}(2k) \text{sinc}\left(\frac{t}{2} - k\right), \quad (2.5)$$

where

$$f_{\text{low}}(2k) = f_{\text{low}}[2k] = \sum_{j \in \mathbb{Z}} h_0[2k - j] f[j]. \quad (2.6)$$

In other words, Equation (2.5) states that it is sufficient for the bandwidth-reduced signal to consider only every second sample.

2.2.2 Ideal High-pass Filter

Conversely to a low-pass filter, a high-pass filter leaves all high frequencies of a signal unchanged, while ‘cutting off’ low frequencies. In analogy to an *ideal* low-pass filter, an *ideal* high-pass filter h_1 is defined as its Fourier transform \hat{h}_1 having no transition between the cut-off frequency and the pass-frequency (see Figure 2.1 (b)):

$$\hat{h}_1(\omega) = \begin{cases} 0 & : \omega \in [-\frac{1}{4}, \frac{1}{4}] \\ 1 & : \omega \in [-\frac{1}{2}, -\frac{1}{4}[\text{ or }]\frac{1}{4}, \frac{1}{2}]. \end{cases}$$

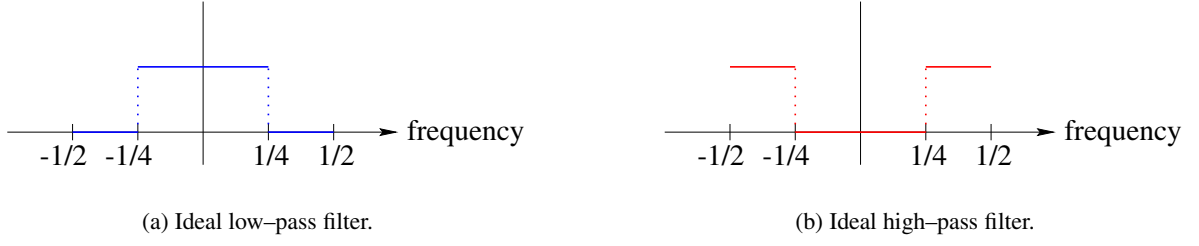


Figure 2.1: Ideal filters.

From the relation

$$\hat{h}_0(\omega) + \hat{h}_1(\omega) = 1 \quad \text{for } \omega \in \left] -\frac{1}{2}, \frac{1}{2} \right]$$

and Equation (2.3) we obtain

$$\begin{aligned} \hat{h}_1(\omega) &= 1 - \left(\frac{1}{2} + \sum_{k \in \mathbb{Z}} \frac{(-1)^k}{(2k+1)\pi} e^{-2\pi i(2k+1)\omega} \right) \\ &= \frac{1}{2} + \sum_{k \in \mathbb{Z}} \frac{(-1)^{k+1}}{(2k+1)\pi} e^{-2\pi i(2k+1)\omega}. \end{aligned} \quad (2.7)$$

It follows that the coefficients of the filter mask are:

$$h_1[0] = \frac{1}{2}, \quad h_1[2k] = 0, \quad h_1[2k+1] = \frac{(-1)^{k+1}}{(2k+1)\pi}. \quad (2.8)$$

The application of a high-pass filter h_1 to f is given by

$$\widehat{h_1 * f}(\omega) = \hat{h}_1(\omega) \hat{f}(\omega) = \hat{f}_{\text{high}}(\omega),$$

and analogous to the calculation in the low-pass filter theory (see Equation (2.4)), we obtain the discrete convolution:

$$f_{\text{high}}[k] = \sum_{j \in \mathbb{Z}} h_1[k-j] f[j]. \quad (2.9)$$

Furthermore, the high-pass filtered function can be expressed with every second sample only:

$$f_{\text{high}}(t) = \sum_{k \in \mathbb{Z}} f_{\text{high}}(2k) \text{sinc}\left(\frac{t}{2} - k\right) \left[2 \cos\left(\pi \left(\frac{x}{2} - k\right)\right) - 1 \right],$$

where (see Equation (2.6))

$$f_{\text{high}}(2k) = f_{\text{high}}[2k] = \sum_{j \in \mathbb{Z}} h_1[2k - j]f[j]. \quad (2.10)$$

The complete signal f can now be completely described as the sum of its low-pass and high-pass filtered parts. With Equations (2.6), (2.10) and the filter masks (2.2) and (2.8), the argumentation of this chapter after some calculation [Dau92] culminates in

$$f(k) = 2 \sum_{j \in \mathbb{Z}} \left[h_0[k - 2j]f_{\text{low}}[2j] + h_1[k - 2j]f_{\text{high}}[2j] \right]. \quad (2.11)$$

Equation (2.11) builds the theoretical foundation for the application of low-pass and high-pass filters in this work. Three important comments conclude the theory of ideal filters:

- The signal f can be written as an infinite sum over high-pass and low-pass filtered copies of itself. Note that the variable j that enters Equation (2.11) is doubled in each iteration step, resulting in a function that is concentrated on half its support. The signal f , which is defined on $[-\frac{1}{2}, \frac{1}{2}]$, is filtered in a first iteration with the low-pass filter on $[-\frac{1}{4}, \frac{1}{4}]$ and a high-pass filter on $[-\frac{1}{2}, -\frac{1}{4}] \cup [\frac{1}{4}, \frac{1}{2}]$. In the next iteration step, the half interval $[-\frac{1}{4}, \frac{1}{4}]$ is further subdivided by a low-pass filter on $[-\frac{1}{8}, \frac{1}{8}]$ and a high-pass filter on $[-\frac{1}{4}, -\frac{1}{8}] \cup [\frac{1}{8}, \frac{1}{4}]$. This process continues for smaller and smaller intervals. The sum of all filter processes gives the original signal f .
- The filter mask that enters the definitions of f_{low} in Equation (2.6), i.e., in the *analysis*, is the same as that in Equation (2.11), i.e., in the *synthesis*. The same holds for h_1 in Equations (2.10) and (2.11).
- The filter masks h_0 and h_1 are shifted by the translation parameter $2j$ for every signal entry k . This means it is sufficient if the convolution of f_{low} with h_0 (respectively, f_{high} with h_1) is executed on every second sample. We have already seen this in the demonstration of the discrete Haar transform in Section 1.7. Equation (2.11), however, states that the *shift* of the convolving filter mask by 2 is a general phenomenon and holds for arbitrary wavelet filters.

For practical purposes, it is desirable to have a representation of the signal f as a *finite* sum of its low-pass and high-pass filtered parts rather than the infinite sum of Equation (2.11). This demand paves the way for the construction of two-channel filter banks whose transition between the pass frequency and the cut-off frequency is no longer ideal.

2.3 Two-Channel Filter Bank

In the notation of filter banks, the above theory of low-pass filters and high-pass filters finds an easy visualization in Figure 2.2 (a).

In applications, the ideal filter bank is not used, as the decomposition of the ideal low-pass filter requires too many (or, more precisely, infinitely many) filter mask coefficients (as can be seen in

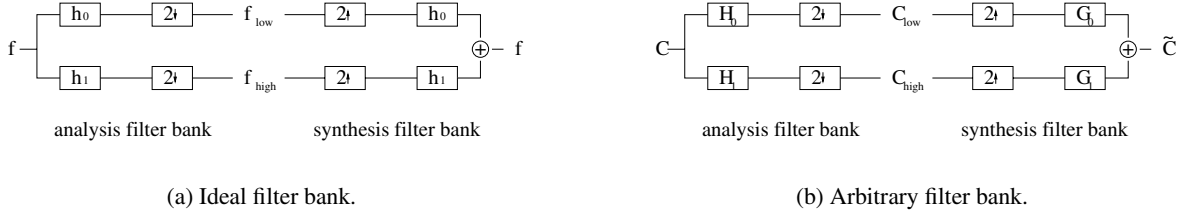


Figure 2.2: Two-channel filter bank, where $2\downarrow$ is the subsampling process, and $2\uparrow$ is the upsampling process. The analysis filter bank decomposes a signal, and the synthesis filter bank reassembles it. (a) Ideal filters: h_0 , respectively, h_1 denote the filter masks in the (discrete) convolution process. (b) Arbitrary filter bank: Filters for analysis and synthesis are not necessarily identical, neither are the input and the output signal.

Equation (2.3)). The same holds for the ideal high-pass filter and Equation (2.7). If the requirements on h_0 and h_1 are released on the condition that a small *transition band* between the cut-off and the pass-frequency is allowed, the Fourier coefficients to describe \hat{h}_0 and \hat{h}_1 decline faster (see Figure 2.3 in comparison to Figure 2.1).

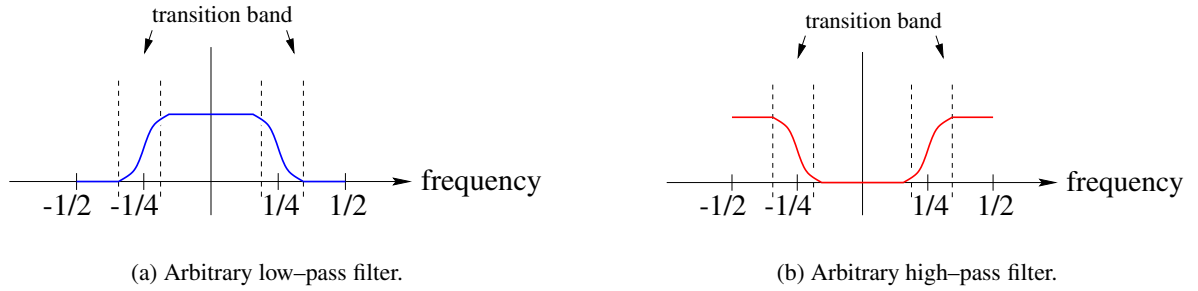


Figure 2.3: Arbitrary low-pass and high-pass filters with transition band between cut-off frequency and pass-frequency.

An arbitrary filter bank is introduced via the z -transform. We will see that the analysis filter mask for an arbitrary filter bank is not identical to the synthesis filter bank. To allow perfect reconstruction of a decomposed signal, we will state conditions on the transfer functions which imply conditions on the filter masks.

Definition 2.1 For a series $(c[k])$ in the vector space l_2 of all converging series $\sum_k |c[k]|^2$, the z -transform is given as

$$C(z) := \sum_{k \in \mathbb{Z}} c[k] z^{-k}$$

with $z \in \mathbb{C}$.

The Fourier series is a special case of this z -transform: With $z = e^{2\pi i \omega}$ we obtain $C(2\pi i \omega) =$

$\sum_{k \in \mathbb{Z}} c[k]e^{-2\pi i k \omega}$. Multiplication and addition of two z -transforms $C(z)$ and $D(z)$ are given by

$$\begin{aligned} D(z)C(z) &= \left(\sum_{m \in \mathbb{Z}} d[m]z^{-m} \right) \left(\sum_{n \in \mathbb{Z}} c[n]z^{-n} \right) = \sum_{k \in \mathbb{Z}} \left(\sum_{j \in \mathbb{Z}} d[j]c[k-j] \right) z^{-k}, \\ C(z) + C(-z) &= 2 \sum_{k \in \mathbb{Z}} c[2k]z^{-2k}, \\ C(z) - C(-z) &= 2z^{-1} \sum_{k \in \mathbb{Z}} c[2k+1]z^{-2k}. \end{aligned}$$

From the arbitrary filter bank in Figure 2.2 (b), we get the following equations on C_{low} , C_{high} and \tilde{C} [Ste00]:

$$\begin{aligned} C_{\text{low}}(z^2) &= \frac{1}{2} \left[C(z)H_0(z) + C(-z)H_0(-z) \right], \\ C_{\text{high}}(z^2) &= \frac{1}{2} \left[C(z)H_1(z) + C(-z)H_1(-z) \right], \\ \tilde{C}(z) &= C_{\text{low}}(z^2)G_0(z) + C_{\text{high}}(z^2)G_1(z), \\ \tilde{C}(-z) &= C_{\text{low}}(z^2)G_0(-z) + C_{\text{high}}(z^2)G_1(-z), \end{aligned}$$

where the first two equations express the analysis and the latter two the synthesis. The synthesized \tilde{C} can thus be expressed in terms of the original signal C as

$$\begin{aligned} \tilde{C}(z) &= \frac{1}{2} \left[H_0(z)G_0(z) + H_1(z)G_1(z) \right] C(z) \\ &\quad + \frac{1}{2} \left[H_0(-z)G_0(z) + H_1(-z)G_1(z) \right] C(-z), \end{aligned} \tag{2.12}$$

where the second term is called an *alias*. We are now looking for conditions on the filter bank to ‘make C and \tilde{C} as similar as possible’. A filter bank with *perfect reconstruction* allows the synthesis to be a multiple of the original, i.e., $\tilde{C}(z) = \alpha C(z)$, or a shift, i.e., $\tilde{C}(z) = z^{-l}C(z)$ with $l \in \mathbb{Z}$.

2.4 Design of Analysis and Synthesis Filters

In order to result in a filter bank of perfect reconstruction, i.e., $\tilde{C}(z) = \alpha z^{-l}C(z)$, we deduce two mathematical conditions on Equation (2.12):

1. The alias term has to vanish, i.e.,

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0. \tag{2.13}$$

2. The synthesis is a shift of the original, i.e.,

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2z^{-l}. \tag{2.14}$$

For simplification, the conditions (2.13) and (2.14) can be written in matrix form. With the setting $-z$ in the above equations, we obtain four conditions:

$$\begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = \begin{bmatrix} 2z^{-l} & 0 \\ 0 & 2(-z)^{-l} \end{bmatrix} \quad (2.15)$$

When the filters are modified so as to result in centered filters, i.e.,

$$G_0^*(z)H_0^*(z) = z^l G_0(z)H_0(z), \text{ respectively, } G_1^*(z)H_1^*(z) = z^l G_1(z)H_1(z),$$

and we agree to use the same notation as above for the centered filters, the right side of Equation (2.15) results in double the identity matrix:

$$\mathbf{G}(z)\mathbf{H}(z) = 2\mathbf{I}. \quad (2.16)$$

This is a very desirable condition since the synthesis filter bank in Equation (2.16) is, besides a factor, the inverse of the analysis filter bank and vice versa.

If we choose

$$\begin{aligned} G_0(z) &= H_1(-z) \\ G_1(z) &= -H_0(-z), \end{aligned} \quad (2.17)$$

the alias condition (2.13) is met. Furthermore, if we replace $H_1(z)$ by $G_0(-z)$ and $G_1(z)$ by $-H_0(-z)$ according to Equation (2.17), the equations of (2.16) are now written differently:

$$\begin{aligned} G_0(z)H_0(z) + G_0(-z)H_0(-z) &= 2 \\ \Leftrightarrow 2 \sum_{n \in \mathbb{Z}} \left[\sum_{k \in \mathbb{Z}} g_0[k]h_0[2n-k] \right] z^{-2n} &= 2 \\ \Leftrightarrow \sum_{n \in \mathbb{Z}} \left[\sum_{k \in \mathbb{Z}} g_0[k]h_0[2n-k] \right] z^{-2n} &= 1 \\ \Leftrightarrow \sum_{k \in \mathbb{Z}} g_0[k]h_0[2n-k] &= \left\langle (g_0[-k]), h_0[k+2n] \right\rangle = \delta_{0n}. \end{aligned} \quad (2.18)$$

Analogous calculations of the equation in (2.16) result in

$$\begin{aligned} \sum_{k \in \mathbb{Z}} g_1[k]h_1[2n-k] &= \delta_{0n}, \\ \sum_{k \in \mathbb{Z}} g_0[k]h_1[2n-k] &= 0, \\ \sum_{k \in \mathbb{Z}} g_1[k]h_0[2n-k] &= 0. \end{aligned}$$

This finally makes explicit the required relationship between (H_0, G_0) , (H_1, G_1) , (H_0, G_1) and (H_1, G_0) for perfect reconstruction. Due to the conditions on (H_0, G_1) and (H_1, G_0) , these filters are called *biorthogonal*.

In the following, we review the construction of analysis and synthesis filters, where the length of the impulse response of the low-pass filter is identical to that of the high-pass filter [Ste00] [Boc98].

2.4.1 Quadrature-Mirror-Filter (QMF)

The definition of the quadrature mirror filters date back to [CEG76].

If the coefficients of the high-pass filter mask h_1 are generated from h_0 by alternating the sign of every second entry, i.e.,

$$\left(\dots, h_1[0], h_1[1], h_1[2], h_1[3], \dots \right) = \left(\dots, h_0[0], -h_0[1], h_0[2], -h_0[3], \dots \right),$$

this is reflected by the system function with the following equations [Boc98]:

$$\begin{aligned} H_1(z) &= H_0(-z) \\ G_0(z) &= H_1(-z) = H_0(z) \\ G_1(z) &= -H_0(-z) = -H_1(z), \end{aligned} \tag{2.19}$$

and the second condition for perfect reconstruction, i.e., the shift condition (2.14) simplifies to

$$H_0^2(z) - H_1^2(z) = 2z^{-l}. \tag{2.20}$$

The name *quadrature-mirror-filter* is due to the fact that $H_1(z) = H_0(-z)$ is the mirror of $H_0(z)$ on the unit sphere $Z = e^{i2\pi\omega}$, and both filters are squared. The condition (2.20), however, cannot be met by finite impulse response (FIR) filters other than the trivial filter of the Haar wavelet.

2.4.2 Conjugate-Quadrature-Filter (CQF)

In this second example, not only the coefficients of the filter mask h_1 are alternated with regard to h_0 , but also the order of the filter coefficients is reversed:

$$\begin{aligned} &\left(\dots, h_1[0], h_1[1], h_1[2], h_1[3], \dots \right) \\ &= \left(\dots, h_0[N], -h_0[N-1], h_0[N-2], -h_0[N-3], \dots \right), \end{aligned} \tag{2.21}$$

with N odd. The system functions are now related by [Boc98]

$$\begin{aligned} H_1(z) &= z^{-N} H_0(-z^{-1}) \\ G_0(z) &= H_0(z^{-1}) \\ G_1(z) &= z H_0(-z) = H_1(z^{-1}). \end{aligned} \tag{2.22}$$

A check of the conditions on perfect reconstruction reveals that Equation (2.13) is satisfied:

$$\begin{aligned}
 & H_0(-z)G_0(z) + H_1(-z)G_1(z) \\
 &= H_0(-z)H_0(z^{-1}) + (-z)^{-1}H_0(z^{-1})zH_0(-z) \\
 &= 0.
 \end{aligned}$$

The second condition (2.14) is not as obvious and has to be met by the choice of H_0 . If we assume $H_0(-1) = 0$, set $z = 1$, and make use of the relation in (2.22), we get

$$\begin{aligned}
 & H_0(z)G_0(z) + H_1(z)G_1(z) \\
 &= H_0(z)H_0(z^{-1}) + z^{-1}H_0(-z^{-1})zH_0(-z) \\
 &= H_0(z)H_0(z^{-1}) + H_0(-z^{-1})H_0(-z) \\
 &\stackrel{z=1}{=} H_0(1)^2 + 0,
 \end{aligned}$$

and condition (2.14) claims that $H_0(1) = \sqrt{2}$ which means

$$\sum_{k \in \mathbb{Z}} h_0[k] = \sqrt{2}, \tag{2.23}$$

i.e., the sum of the low-pass filter coefficients is $\sqrt{2}$. Indeed, this two-channel filter bank with perfect reconstruction realizes an orthogonal decomposition.

These CQF filters will be the basis for our research on suitable parameter settings for still image and layered video coding presented in Sections 6.3 and 7.5. For the implementation, we have mainly concentrated on the Daubechies wavelets [Dau92] [Mal98]. The implementation of the wavelet-based audio denoiser, however, relies on other CQF filters as well.

Chapter 3

Practical Considerations for the Use of Wavelets

One man's hack is another man's thesis.
– Sean Levy

3.1 Introduction

The previous sections concentrated on the theory of wavelets: definition of the continuous wavelet transform, comparison of the time–frequency resolution of the short–time Fourier and the wavelet transforms, and the multiscale theory, which paved the way for the fast (discrete, dyadic) wavelet transform. Finally, the general context of filter banks and the wavelet filter design was elaborated.

This chapter addresses practical considerations for the use of wavelets. The topics encompass the *extension of wavelet filters* into multiple dimensions and the induced different *decomposition policies*, various *boundary extension policies*, the problem of *visualization of the coefficients in the time–scale domain*, and *decoding policies* in a scenario in which the decoder has not yet received the full information due to compression or network reasons. Finally, it gives a brief introduction to the idea of implementing the wavelet transform via the *lifting scheme*, which is used in the JPEG2000 standard.

The discussion of this practical tool kit for implementations is our own contribution to the wavelet theory. It has been presented at [Sch01d].

3.2 Wavelets in Multiple Dimensions

The fast wavelet transform introduced in Section 1.5 culminates in Equation (1.20), which presents the decomposition of the function space, where the original signal ‘lives’, into approximation function spaces and detail function spaces. This multiresolution theory is ‘per se’ defined only on one–dimensional signals. Application of the wavelet transform on still images and video requires an ex–

tension into multiple dimensions.

Two- and three-dimensional wavelet filter design is an active field of research which we briefly outline in Section 3.2.1. In Section 3.2.2 we present the current implementations, the different decomposition policies they allow, and their interpretations.

3.2.1 Nonseparability

Separability denotes the fact that the successive application of a one-dimensional filter into one dimension and afterwards into the second dimension is mathematically identical to a two-dimensional wavelet transform from the outset.

Nonseparable wavelet filter banks in two or three dimensions are very desirable as they incorporate the *real* idea of multiple dimensions. This means an image — as example of a two-dimensional signal — will be treated on the two-dimensional plane. The ‘true’ multidimensional case means that both nonseparable sampling and filtering are allowed. Although the nonseparable approach generally suffers from higher computational complexity, it would also overcome the drawbacks of separability and provide:

- more freedom in filter design,
- less stress on the *horizontal*, *vertical*, and *diagonal* orientations, and
- schemes that are better adapted to the human visual system.

With nonseparable filters, the notion of a *sampling lattice* enters the discussion. Nonseparable wavelet filters have been researched by the groups around Kovačević [KV92] [KV95], Vetterli [KVK88], and Tay [TK93]. They have not been considered in this work as their design has remained mathematical and has not yet been applied.

3.2.2 Separability

The discussion of separability is presented for the two-dimensional case. Once this concept is understood, an extension into multiple dimensions is trivial. For simplification, we use a different type of indexing than before: instead of discussing the approximation levels 2^j , we will consider the levels k with $k \in \mathbb{Z}$.

The successive convolution of filter and signal in both dimensions opens two potential iterations: *nonstandard* and *standard* decompositions. When Equation (1.7) is extended into two dimensions via the tensor product, i.e., $V_0^{(2)} = V_0 \times V_0$, the decomposition into approximation and detail starts identically in the first decomposition step:

$$\begin{aligned}
 V_0^{(2)} &= V_0 \times V_0 \\
 &= (V_1 \oplus W_1) \times (V_1 \oplus W_1) \\
 &= V_1 \times V_1 \oplus V_1 \times W_1 \oplus W_1 \times V_1 \oplus W_1 \times W_1
 \end{aligned} \tag{3.1}$$

$$=: (\square).$$

Equation (3.1) indicates that the two-dimensional wavelet analysis splits an image into sub-parts of different scales. At each level of iteration, structures of a specific frequency range are separated out. Moreover, the details of each split-off scale are further subdivided into specific orientations: Let

- $V_1 \times V_1$ be spanned by the basis functions $\{\varphi_k^1 \varphi_l^1\}_{k,l} =: \{\varphi_{kl}^1\}$,
- $V_1 \times W_1$ be spanned by $\{\varphi_k^1 \psi_l^1\}_{k,l} =: \{\psi_{kl}^{1,\text{hor}}\}$,
- $W_1 \times V_1$ be spanned by $\{\psi_k^1 \varphi_l^1\}_{k,l} =: \{\psi_{kl}^{1,\text{ver}}\}$,
- $W_1 \times W_1$ be spanned by $\{\psi_k^1 \psi_l^1\}_{k,l} =: \{\psi_{kl}^{1,\text{diag}}\}$.

Then each summand will process the x -axis first, followed by the y -axis. In total, the separable two-dimensional wavelet transform privileges the angles of 0° , 45° , and 90° . The critically sampled discrete wavelet transform of Figures 3.1 (a) and (b) shrinks the detail images and the approximation in each iteration step to one fourth the original size, and the total amount of coefficients remains constant¹.

3.2.2.1 Standard Decomposition

When Equation (3.1) is iterated, the standard decomposition iterates on *all* approximation spaces V_1 , resulting in

$$\begin{aligned} (\square) &= (V_2 \oplus W_2) \times (V_2 \oplus W_2) \oplus (V_2 \oplus W_2) \times W_1 \\ &\quad \oplus W_1 \times (V_2 \oplus W_2) \oplus W_1 \times W_1 \end{aligned} \quad (3.2)$$

$$\begin{aligned} &= V_2 \times V_2 \oplus V_2 \times W_2 \oplus W_2 \times V_2 \oplus W_2 \times W_2 \\ &\quad \oplus V_2 \times W_1 \oplus W_2 \times W_1 \oplus W_1 \times V_2 \\ &\quad \oplus W_1 \times W_2 \oplus W_1 \times W_1 \end{aligned} \quad (3.3)$$

after the second iteration step, thus in nine summands. In this sum, the only remnants of the first iteration (see Equation (3.1)) are the details of step 1, i.e., $W_1 \times W_1$. The approximations V_1 of the first iteration are dissected into approximations and details of the next level, i.e., V_2 and W_2 .

3.2.2.2 Nonstandard Decomposition

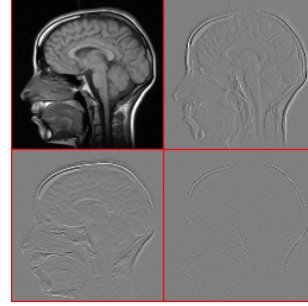
The nonstandard decomposition iterates only the *purely* low-pass filtered approximations $V_1 \times V_1$. This results in

$$(\square) = (V_2 \oplus W_2) \times (V_2 \oplus W_2) \oplus V_1 \times W_1$$

¹However, since grayscale images are typically represented by 1-byte integers, while the coefficients in the wavelet-transformed time-scale domain are real values, the storage space actually expands. The same holds true for color images.

$V_1 \times V_1$	$V_1 \times W_1$
$W_1 \times V_1$	$W_1 \times W_1$

(a) Start of iteration; idea.



(b) Start of iteration; visualization.

(c) Standard decomposition.

(d) Nonstandard decomposition.

$\psi_{kl}^{1,diag}$	$\psi_{kl}^{1,ver}$	$\psi_{kl}^{1,diag}$
$\psi_{kl}^{1,hor}$	φ_{kl}^1	$\psi_{kl}^{1,hor}$
$\psi_{kl}^{1,diag}$	$\psi_{kl}^{1,ver}$	$\psi_{kl}^{1,diag}$

(e) Frequency localization (i.e., Fourier domain) for one iteration.

(f) Frequency localization (i.e., Fourier domain) for three iterations. Solid lines: nonstandard. Solid and dotted lines: standard.

Figure 3.1: Separable wavelet transform in two dimensions. (a) visualizes the four summands of Equation (3.1). (b) presents the first iteration of the image *Brain* by the Haar wavelet. It also demonstrates that the upper right part contains the details in the horizontal direction, while the lower left part contains the details in the vertical direction. The lower right corner contains diagonal details. (c) and (d) visualize the two methods of decomposition after 4 iterations. (e) and (f) present the frequency localization on the Fourier plane, achieved by multiresolution analysis.

$$\begin{aligned}
& \oplus W_1 \times V_1 \oplus W_1 \times W_1 \\
= & V_2 \times V_2 \oplus V_2 \times W_2 \oplus W_2 \times V_2 \oplus W_2 \times W_2 \\
& \oplus V_1 \times W_1 \oplus W_1 \times V_1 \oplus W_1 \times W_1,
\end{aligned} \tag{3.4}$$

thus in seven summands after the second iteration step. In this nonstandard decomposition, the mixed terms $V_1 \times W_1$ and $W_1 \times V_1$ of the first iteration remain unchanged.

Thus the difference between the two decompositions is that the standard decomposition iterates also the parts of the approximations that are located within mixed terms, while the nonstandard decomposition iterates only purely low-pass filtered components. Consequently, the standard decomposition results in many more summands in the time-scale domain. Figures 3.1 (c) and (d) demonstrate the two policies in graphical form for four iteration steps.

The frequency localization of the nonstandard decomposition in the Fourier domain is presented in Figures 3.1 (e) and (f). In Section 2.3, we have discussed that the ideal frequency separation remains a theoretical case. Due to implementation issues, the exact separation of the frequency bands in Figures 3.1 (e) and (f) does not exist, and actually, the separating lines and dotted lines should be blurred.

For arbitrary dimensions, the concept that we have discussed here for two dimensions is simply extended.

3.3 Signal Boundary

A digital filter is applied to a signal by convolution (see Section 2.3 and Equations (2.4) and (2.9)). Convolution, however, is defined only *within* a signal. In order to result in a mathematically correct, reversible wavelet transform, *each* signal coefficient must enter into $\text{filter_length}/2$ calculations of convolution (here, the subsampling process by factor 2 is already incorporated). Consequently, every filter longer than 2 coefficients or *taps*, i.e., every filter except *Haar*, requires a solution for the boundary coefficients of the signal.

The boundary treatment becomes even more important the shorter the analyzed signal is. An audio piece, considered as a one-dimensional discrete signal over time and at a sampling rate of 44.100 samples per second contains so many coefficients in its interior that — independent of its actual length — the boundary plays only a subordinate role. Still images, however, are signals of a relatively short length (in rows and columns), thus the boundary treatment is very important. Two common boundary policies are *circular convolution* and *padding*.

3.3.1 Circular Convolution

The idea of circular convolution is to ‘wrap’ the signal around at the boundary, i.e., wrap the end of a signal to the beginning (or vice versa). Figure 3.2 (a) illustrates this approach with a signal of length 8 and a filter of 6 taps. The convolution of the signal entries 1 to 6 with the filter results in the entry a in the time-scale domain. In the same manner, the convolution of the signal entries 3 to 8 with

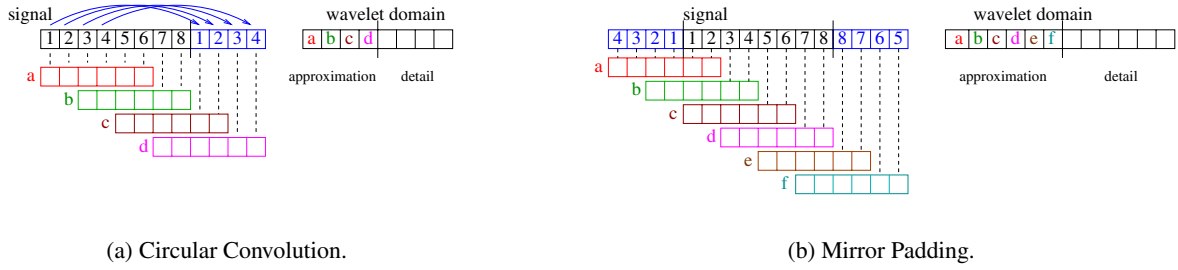


Figure 3.2: Circular convolution versus mirror padding for a signal of length 8 and a filter of 6 taps. Here, the filter is a low-pass filter, thus the coefficients resulting from the convolution form the approximation entries. In (a), the approximation contains half as many entries as the original signal. Together with the details, the entries in the wavelet domain require the same storage space as the original signal. In (b), the padding results in inflated storage space in the wavelet domain.

the filter results in the entry b in the time-scale domain. The process has not finished yet, but for the next convolution, the signal entries 5 to 8 are not enough and two more signal entries are required. Furthermore, the entries 1 and 2 need to be included in $\text{filter_length}/2$, i.e., in 3 calculations. Thus, they are being ‘wrapped’ around and enter the calculation of the time-scale coefficient c . The same is done with d .

Figure 3.2 (a) demonstrates the convolution with a low-pass filter. The number of approximation coefficients is only half as many. A convolution with a high-pass filter would produce an equal number of detail coefficients.

In so doing, circular convolution is the only boundary treatment that maintains the number of coefficients for a wavelet transform, thus simplifying storage handling. However, the time information contained in the time-scale domain of the wavelet transformed coefficients ‘blurs’: The coefficients in the time-scale domain that are next to the right border (respectively, left border) also affect signal coefficients that are located on the left (respectively, on the right). In the example in Figure 3.2 (a) this means that information on pixels 1 and 2 of the left border of the original signal is contained not only in entries of the time-scale domain that are located on the left, but also on the right side, i.e., in the entries a , c and d of the time-scale domain. c and d are the coefficients that, due to circular convolution, contain information from the ‘other’ side of the signal.

3.3.2 Padding Policies

Padding policies have in common that they add coefficients to the signal on either of its borders. The border pixels of the signal are padded with $\text{filter_length}-2$ coefficients. Consequently, each signal coefficient enters into $\text{filter_length}/2$ calculations of convolution, and the transform is reversible. Many padding policies exist: *zero padding*, where 0’s are added, *constant padding*, where the signal’s boundary coefficient is padded, *mirror padding*, where the signal is mirrored at the boundary, and *spline padding*, where the last n_0 border coefficients are extended by spline interpolation, etc.

All padding policies have in common that storage space in the wavelet domain is physically enlarged

by each iteration step, see Figure 3.2 (b). Though the amount of storage space required can be calculated in advance (see [Wic98]), it nevertheless remains sophisticated: Storage handling is strongly affected by the fact that only the *iterated* parts expand, thus expansion depends on the selected decomposition policy (see Section 3.2.2) and on the iteration level.

A strength of all padding approaches, however, is that the time information is preserved. This means that signal coefficients that are located on the left (respectively, on the right) are represented by time–scale coefficients at the same location.

3.3.3 Iteration Behavior

Convolving the signal with a filter is reasonable only for a signal length greater than the filter length, and each iteration step reduces the size of the approximating signal by a factor of 2. This does not affect the iteration behavior of padding policies.

In circular convolution, however, the decomposition depth varies with the filter length: the longer the filter, the fewer decomposition iterations are possible. For an image of 256×256 pixels, the Daubechies–2 filter bank with 4 taps allows a decomposition depth of seven levels, while the Daubechies–20 filter bank with 40 taps reaches signal length after only three decomposition levels. Table 3.1 gives some more iteration levels for circular convolution.

Filter Bank	Taps	Iterations
Daubechies–2	4	7
Daubechies–3	6	6
Daubechies–4	8	6
Daubechies–5	10	5
Daubechies–10	20	4
Daubechies–15	30	4
Daubechies–20	40	3

Table 3.1: The number of possible iterations on the approximation part of an image of 256×256 pixels when circular convolution is applied, depends on the length of the filter bank.

Depending on the selected quantization policy for compression, the number of iterations can strongly affect the quality of a decoded signal. This is discussed in detail in Section 6.3 in the context of parameter evaluation for image coding, and in Section 7.5 in the context of video layering policies for hierarchical coding.

3.4 ‘Painting’ the Time–scale Domain

So far, we have discussed the wavelet analysis, i.e., the decomposition of a signal into its coefficients in the time–scale domain. This section discusses the challenge to ‘paint’, i.e., visualize the coefficients in the wavelet–transformed time–scale domain.

Since the multiresolution aspect of the wavelet transform allows a direct interpretation of the time-scale domain as approximation and details (see Section 1.6), the coefficients in the transformation-space can be visualized. This is especially suited for still images since it gives a more intuitive meaning to the coefficients in the time-scale domain and facilitates their interpretation. However, the wavelet-transformed coefficients are *not* pixel values, and different aspects need to be considered when visualizing them: *normalization* and *range*.

3.4.1 Normalization

The orthogonal wavelet filter banks with perfect reconstruction discussed in this work (see Section 2.4 and Equation (2.23)) have the property that the sum of the low-pass filter coefficients is $\sqrt{2} > 1$. Application of this filter to a signal thus raises the average luminance by the factor $\sqrt{2}$. Pixel values, however, can be painted only in the discrete range of 0 (black) to 255 (white).

One solution is to set all pixel values in the time-scale domain brighter than 255 to 255 (see Figure 3.3 (a), approximation part (left)). Similarly, the high-pass filter results in detail information towards the approximation. In other words, the details specify the variation of a specific pixel towards an average. This variation can be positive or negative. One could draw these coefficients by cutting off the negative parts and considering only the positive values (Figure 3.3 (a), detail part (right)). Iteration of the wavelet transform on this visualization results in an approximation part of an image that grows brighter and brighter, while the detail parts remain mostly black.

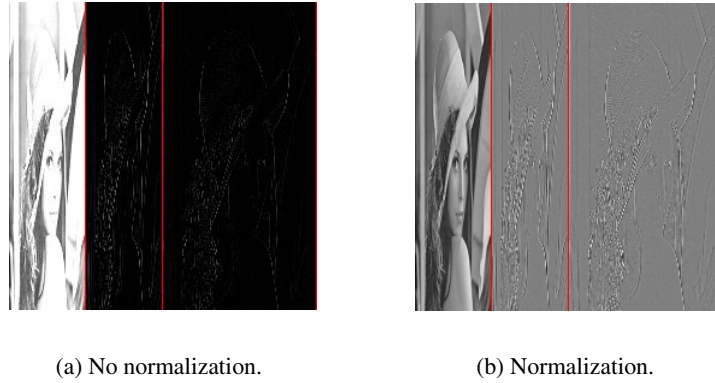


Figure 3.3: Two possible realizations of ‘painting the time-scale coefficients’ (here: Daubechies-2 wavelet filter and standard decomposition at level 2).

With the term *normalization*, we denote the fact that the coefficients in the time-scale domain are ‘edited’ before they are visualized. This means that the coefficients in the low-pass filtered regions are divided by powers of $\sqrt{2}$ prior to visualization. This keeps the total luminance of the approximation constant. The high-pass filtered coefficients are elevated by 128, so that the detail parts have an average value of mid-gray, while the former negative variations appear to the darker and former positive variations appear to the brighter (see Figure 3.3 (b)).

This process, however, requires *two* copies of the coefficients in the time-scale domain: one copy for

the visual representation and the second for the calculation of the coding process.

3.4.2 Growing Spatial Range with Padding

As we have discussed in Section 3.3.2, boundary padding policies result in an enlarged time–scale domain. This enlargement increases with every iteration. Moreover, only the iterated (i.e., low–pass filtered) parts are inflated, thus the time–scale domain does not grow symmetrically.

We illustrate the problem by an example. We analyze an image of size 256×256 pixels with the Haar filter bank (2 taps) and the Daubechies–20 filter bank (40 taps). The decomposition policy is nonstandard, the boundary is treated with zero padding. Table 3.2 shows the size of the approximation (i.e., purely low–pass filtered ‘upper left corner’) at different levels of iteration.

Level of iteration	Size of ‘upper left corner’	
	Haar filter	Daub–20 filter
1	128×128	147×147
2	64×64	93×93
3	32×32	66×66
4	16×16	52×52
5	8×8	45×45
6	4×4	42×42
7	2×2	40×40
8	1×1	39×39

Table 3.2: The size of the time–scale domain with padding depends on the selected wavelet filter bank.

Consequently, the coefficients in the time–scale domain in the example with the Daubechies–20 filter contain many ‘padded’ coefficients, and only a minor number of ‘real’ approximation coefficients. When the time–scale domain of a wavelet–transformed image with padding policy is visualized, one actually ‘cheats’ a bit, as one cuts off the padded coefficients from visualization. Figure 3.4 illustrates the problem.

This raises a new question: *How can we distinguish the ‘real’, i.e., approximating, coefficients in the time–scale domain from the padding coefficients?* The number of ‘real’ approximation coefficients at each level is known. In the implementation of the Java applet for the wavelet transform on still images (see Section 9.6), the method of finding them has been realized differently for each type of padding:

- With zero padding, the implementation supposes that the original image is not all black. An iteration on the rows and columns of the image then searches for non–black boundary pixels (see Figure 3.4 (a)). As the target size of the ‘real’ approximation is known, this approach is stable even if some black border pixels are present.
- Mirror padding does not allow the same easy approach. Figure 3.4 (b) illustrates that the low–pass filtered coefficients in the time–scale domain with mirror padding extend in each iteration with mirrors of the image’s borders. These have the same gray values as the original image, however; thus the approximation signal cannot be detected by comparison of the gray values to

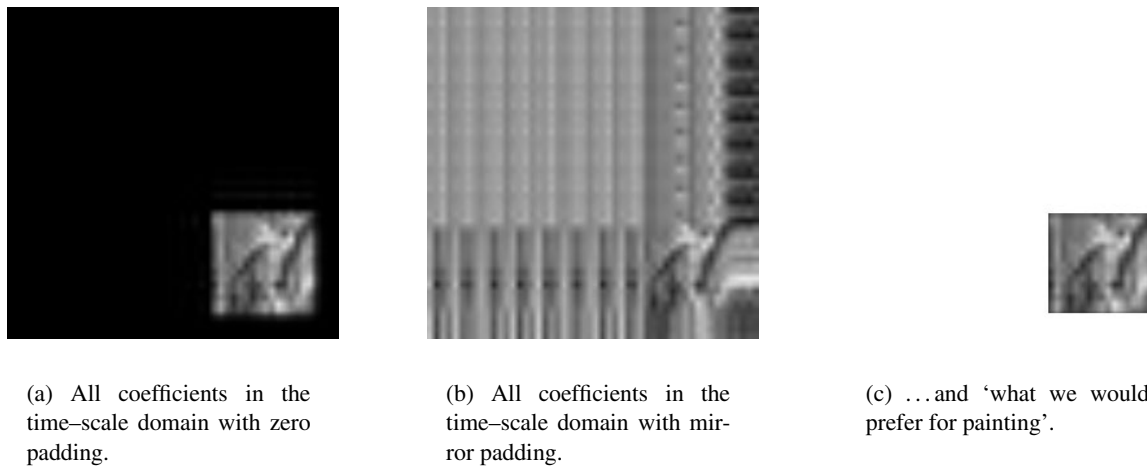


Figure 3.4: Trimming the approximation by zero padding and mirror padding. The parameters have been set to nonstandard decomposition, Daubechies–20 wavelet filter bank and iteration level 4.

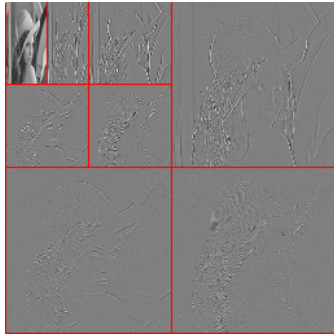
the ‘padded’ coefficients. Our solution was to cut out a piece of the target size from the center of the corresponding time–scale domain. As the ‘real’ approximations are not necessarily in the center (see Figure 3.4), this approach is unstable, i.e., the deep iteration steps might ‘draw’ coefficients in the low–pass filtered parts of the image that signify padding rather than ‘real approximation’.

3.5 Representation of ‘Synthesis–in–progress’

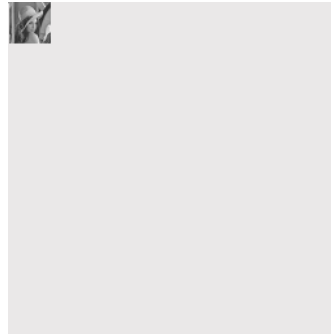
The Internet is the most common source to find or distribute wavelet–encoded multimedia data. In this section, we discuss different policies to visually represent a decoded image when the decoder has not yet received the complete information due to compression or transmission delays (e.g., in the Internet environment). Like in progressive JPEG [PM93], wavelet–transformed data can be represented when synthesis is still in progress.

The synthesis of an encoded image begins with the approximation part in the time–scale domain and subsequently adds information contained in the band–pass and high–pass filtered regions of the time–scale domain, which increases spatial resolution. There are three ways to represent the subsequent resolution of an encoded signal: A synthesis–in–progress can be represented by *reversal of the analysis*, by *growing spatial resolution* or by *interpolation*.

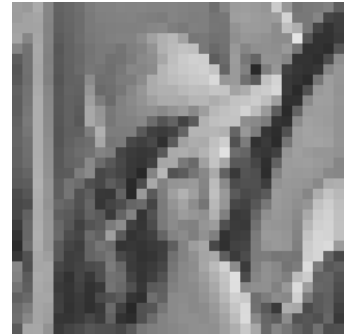
Analysis Reversal is the canonical method of visualization, where the synthesized image ‘grows blockwise’. Figure 3.5 (a) shows the decoding process once the vertical details of level 3 have already been decoded, and thus added to the approximation part, but the horizontal details have not (thus level ‘2.5’).



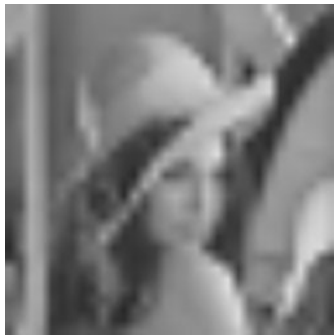
(a) Analysis reversal.



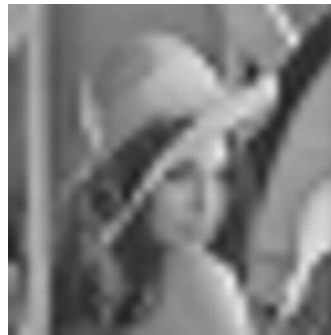
(b) Growing spatial resolution.



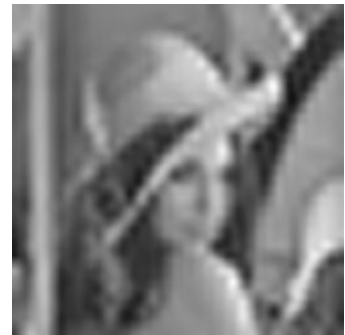
(c) Interpolation, block wise.



(d) Interpolation, bilinear.



(e) Interpolation, cubic.



(f) Interpolation, bicubic.

Figure 3.5: Representation of synthesis-in-progress for the 256×256 grayscale image *Lena*. The image is analyzed using the Daubechies-2 wavelet filter, nonstandard decomposition of depth 7, and circular convolution. (a) Analysis reversal at level ‘2.5’. All images (b) — (f) contain the same amount of information in the time-scale domain, i.e., the approximation of 32×32 pixels in level 3. (b) demonstrates the ‘growing spatial resolution’ while (c) — (f) expand the approximation to original size using different interpolation algorithms.

Growing spatial resolution ‘draws’ only the purely low-pass filtered approximation. When the synthesis starts, the approximation is a very small image (in the extreme, 1×1 pixel, depending on the parameters). Subsequently, as more and more information is added, the spatial size of this approximation continues to grow until it has reached the size of the original (see Figure 3.5 (b)). This approach implements growth in the form of the Laplacian pyramid [BA83]. It is similar to the *analysis reversal* mentioned above. However, only *fully* decoded iteration levels are painted. Thus, it is a coarser representation than the analysis reversal.

Interpolation always inflates the current approximation to the original size of the image and adds missing pixels by means of interpolation (see Figures 3.5 (c) — (f)). The question remains which interpolation technique shall be implemented: simple blockwise ‘cloning’ (c), linear interpolation, bilinear (d), cubic (e), or bicubic (f) — there are many options. In general, visual results are acceptable after cubic interpolation.

3.6 Lifting

A different technique to construct biorthogonal wavelets and multiresolution has been introduced by Sweldens [Swe96] [DS98]. His approach is called the *lifting scheme* or *second generation wavelets*. The main difference to the classical construction presented in Section 2.4 is that lifting does not rely on the Fourier transform (see Section 1.3 and Equation (1.1)). This can be used to construct wavelets which are not necessarily translates and dilates of one *mother wavelet* like that in Section 1.5.

The lifting scheme has the following advantages compared to the construction and implementation discussed so far [Swe96]:

1. It allows a faster implementation of the wavelet transform as it makes optimal use of similarities between high- and low-pass filters. The number of floating point operations can be reduced by a factor of two (see the analysis part of Figure 2.2 and compare it to Figure 3.6).
2. It allows a fully *in-place calculation* of the wavelet transform. This means, in contrast to what has been discussed in Section 3.3, no auxiliary memory is needed, and the original signal can be replaced by its wavelet transform.

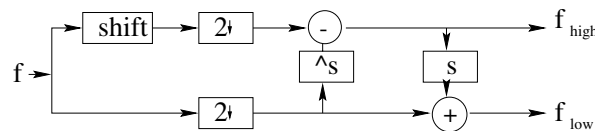


Figure 3.6: Analysis filter bank for the fast wavelet transform with lifting.

Sweldens also states that the lifting scheme is a didactically valuable transform as it is immediately clear that the synthesis is the inverse of the analysis.

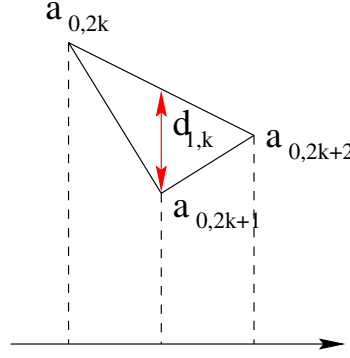


Figure 3.7: Lifting scheme: prediction for the odd coefficients as difference from the linear approximation.

Let us look at an example to motivate this new approach. A signal a_0 with sampling distance 1 shall be decorrelated, just as it was decorrelated with the Haar transform (see Section 1.7). By simply subsampling the even samples of the original signal, we obtain a new sequence of *approximations*

$$a_{1,k} := a_{0,2k} \quad \text{for } k \in \mathbb{Z}. \quad (3.5)$$

A trivial way to capture the lost information is to say that the *detail* is simply contained in the odd samples, $d_{1,k} = a_{0,2k+1}$. A more elaborate way is to recover the original samples from the subsampled coefficients $a_{1,k}$. The even samples of the original signal then are immediately obvious. The odd samples, however, are the difference between the linear prediction of the two neighboring even samples, and the odd sample (see Figure 3.7):

$$d_{1,k} := a_{0,2k+1} - \frac{1}{2}(a_{0,2k} + a_{0,2k+2}). \quad (3.6)$$

These details, considered as wavelet coefficients, essentially measure the extent to which the original signal fails to be linear, and their expected value is small. In principle, we could now iterate this scheme. However, the choice of the approximation coefficients $a_{1,k}$ could still be improved: In the definition of Equation (3.5), the even samples at positions 2^j stay constant within j iterations. This introduces considerable aliasing. In order to preserve the average value of all coefficients at each level j , i.e., $2 \sum_k a_{j+1,k} = \sum_k a_{j,k}$, the approximations $a_{1,k}$ are *lifted* again with the help of the wavelet coefficients $d_{1,k}$. This yields [Swe96]:

$$a_{1,k} = a_{0,2k} + \frac{1}{4}(d_{1,k-1} + d_{1,k}). \quad (3.7)$$

The wavelet transform at each level now consists of two stages: The calculation of the wavelet coefficients as the difference to the linear prediction (Equation (3.6)) and the lifting of the approximations by these details (Equation (3.7)). This scheme is demonstrated in Figure 3.8.

The synthesis of this lifting-based analysis is the simple reverse of Equations (3.7) and (3.6).

If this sample lifting-based transform is again expressed as high-pass and low-pass filters, we see that a detail coefficient is influenced by three signal coefficients of the next-finer level, while an

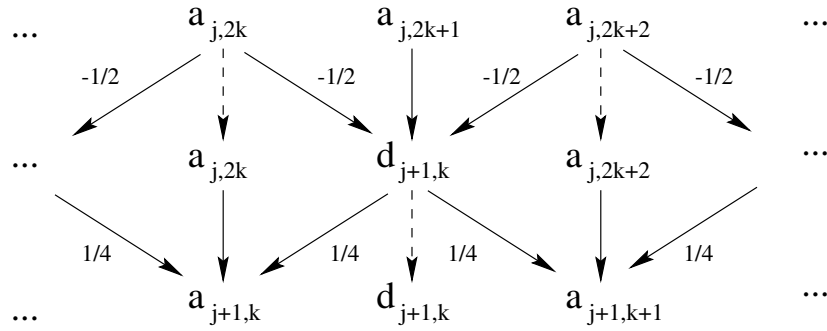


Figure 3.8: The lifting scheme. This figure visualizes the computation of Equations (3.6) and (3.7). Upper row: signal samples at level j . Middle row: computation of details at level $j+1$ while the approximations are not yet treated. Lower row: computation of approximations at level $j+1$ based on the heretofore computed details.

approximation coefficient is influenced by five coefficients of the next-finer level. The high-pass filter entries are immediately obvious from Equation (3.6) and correspond to $[-\frac{1}{2} \ 1 \ -\frac{1}{2}]$. The low-pass filter entries are calculated as follows: The influence of $a_{j,2k}$ on $a_{j+1,k}$ is $1 + (-\frac{1}{2})\frac{1}{4} + (-\frac{1}{2})\frac{1}{4} = \frac{6}{8}$ (see also Figure 3.8). The influence of both $a_{j,2k-1}$ and $a_{j,2k+1}$ is $\frac{1}{4}$, and both $a_{j,2k-2}$ and $a_{j,2k+2}$ enter $a_{j+1,k}$ with the factor $-\frac{1}{2} \cdot \frac{1}{4} = -\frac{1}{8}$. Thus, the low-pass filter mask is $[-\frac{1}{8} \ \frac{2}{8} \ \frac{6}{8} \ \frac{2}{8} \ -\frac{1}{8}]$.

The filters we have just derived are the default *reversible wavelet transform* Daub-5/3 filters suggested in the standard JPEG2000 [SCE00a] [ITU00]. In the context of JPEG2000, an *irreversible* wavelet filter bank is defined as well, denoted as Daub-9/7.

The irreversible wavelet transform iterates the presented lifting scheme. That is, not only are the details of the next iteration level $j+1$ calculated based on the approximations of the current level j , and the approximations of the next iteration are calculated based on these details at level $j+1$ (see Figure 3.8). In contrast, the computation of the details described above is weighted by parameters, as is the computation of the approximations; both are only *intermediate* values. The computation of the details of level $j+1$ then is based on these intermediate approximations, and so the computation of the approximations relies on these final details at level $j+1$. The influence of the coefficients of level j on the coefficients of level $j+1$ thus is by summand 4 more widespread: The details at level $j+1$ are based on 7 (rather than 4) original samples at level j , while the approximations are based on 9 (rather than 5) approximations at level j .

The coefficients of both filter banks are given in Table 3.3. In contrast to the coefficients of the *reversible* filter bank which allow infinite precision through their computation as finite fractions, the notion *irreversible* has been attached to the Daub-9/7 filter bank, because due to the real-valued parameters that enter the filter computation, its coefficients are rounded values. The JPEG2000 standard will be briefly outlined in Section 6.4.1 where the focus will be on the coding of specific important regions within still images.

This concludes our discussion of the theory of wavelets and practical considerations concerning its use in audio and image processing. We have introduced the important notion of time-frequency analysis and the major advantage of wavelet analysis compared to the short-time Fourier analysis. Chapter 3 has outlined challenges that spring up in concrete implementation tasks. Answers to questions such

Daub-5/3 Analysis and Synthesis Filter Coefficients				
i	Analysis Filter		Synthesis Filter	
	low-pass	high-pass	low-pass	high-pass
0	6/8	1	1	6/8
± 1	2/8	-1/2	1/2	-2/8
± 2	-1/8			-1/8

Daub-9/7 Analysis and Synthesis Filter Coefficients				
i	Analysis Filter		Synthesis Filter	
	low-pass	high-pass	low-pass	high-pass
0	0.6029490182363579	1.115087052456994	1.115087052456994	0.6029490182363579
± 1	0.2668641184428723	-0.5912717631142470	0.5912717631142470	-0.2668641184428723
± 2	-0.07822326652898785	-0.05754352622849957	-0.05754352622849957	-0.07822326652898785
± 3	-0.01686411844287495	0.09127176311424948	-0.09127176311424948	0.01686411844287495
± 4	0.02674875741080976			0.02674875741080976

Table 3.3: Filter coefficients of the two default wavelet filter banks of JPEG2000.

as the definition of: the choice of a suitable wavelet filter bank, the treatment of a signal's boundary, the level of iterations on the approximation part of a signal, and the question of how to represent a decoded signal when the receiver has not yet obtained the complete information, however, depend on the underlying signal as well as on the specific task.

In Part II, we enter the applications of audio, image and video coding and present promising applications of wavelet transform coding.

Part II

Application of Wavelets in Multimedia

Chapter 4

Multimedia Fundamentals

The real danger is not that computers will begin to think like men, but that men will begin to think like computers.

– Sydney J. Harris

4.1 Introduction

Part I of this book discussed the theoretical aspects as well as the practical considerations of the wavelet transform. Part II employs these concepts and seeks novel applications for multimedia coding.

One definition of the notion *multimedia* often referred to in the literature is the following:

Multimedia signifies the processing and integrated presentation of information in more than one form, e.g., as text, audio, music, images, and video.

In the context of this dissertation, the term multimedia refers to the three signal processing subgroups *audio*, *still images*, and *video*, where the notion of *signal processing* summarizes all techniques to analyze and modify a signal. Figure 4.1 shows a typical processing system with an analog input (e.g., speech), an analog to digital conversion, a digital processor which forms the heart of the process, the re-conversion of the digital signal into an analog one and finally the analog output.

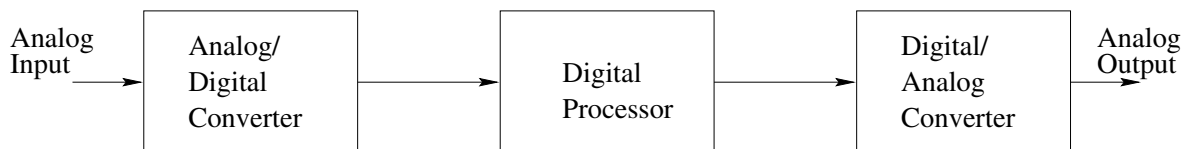


Figure 4.1: Digital signal processing system.

The digital processor can have either of the two main purposes *analysis* of the signal, i.e., the decomposition into its building components, extraction, and manipulation of certain interesting features, or *compression*, i.e., the reduction of storage space. Both applications are correlated since compression

ideally works to minimize the perceptible loss of quality; this goes along with analysis of the signal and maintenance of the most important characteristics. In Chapter 5 we will see that our main audio application entails *analysis*, while both image and video coding focus on *compression* (see Chapters 6 and 7).

This chapter on multimedia fundamentals introduces the concept of data compression in Section 4.2 and classifies different aspects of importance in terms of the underlying problem. Fundamental to digital audio processing (see Chapter 5) is its implication that a *digital* system can be designed which does not lose any of the information contained in its *analog* counterpart. In Section 4.3, the theory of sampling is briefly reviewed.

4.2 Data Compression

Data compression is both the art and the science of reducing the number of bits required to describe a signal [PM93]. Techniques are classified primarily as either *lossless* and *lossy* compression techniques [Ohm95]. A combinations of both approaches is referred to as *hybrid* coding [ES98]. Thus three compression categories exist:

- *Entropy Coding* is a lossless compression technique. The notion of entropy has emerged in thermodynamics: If a thermodynamic system (or a data source) is well-organized and contains only little haphazardness, then the amount of entropy is small [Str00]. A large amount of entropy in contrast denotes a state of great disorder. In information technology, the largest possible amount of entropy means an equal probability distribution over the complete code alphabet.

Entropy coding comprises *run-length coding*, *pattern substitution*, *Huffman coding*, *arithmetic coding*, etc. It realizes a clever treatment of the data as it searches for redundancies and a realistic probability distribution in order to minimize storage space. The decompressed data are identical to the original.

- *Source Coding*, by contrast, is a lossy process. It takes advantage of the fact that the data are destined for the human as the data sink. The human visual and auditory systems are crucial to source coding as this approach exploits their deficiencies in order to discard information imperceptible to the human ear or eye.

Source coding techniques comprise *interpolation* and *subsampling*, *fractal coding*, and all *transform-based coding techniques* such as the discrete cosine and wavelet transforms. More precisely, the transform itself implies no loss of data, or else only a minor one due to arithmetic rounding operations by the computer. But a subsequent *quantization* of the transformed data discards information, so that the process is not reversible (see also the quotation from Daubechies in Section 2.1).

- Most compression standards combine both coding techniques into so-called *hybrid coding*. They first transform and quantize the original signal (i.e., perform source coding) and then entropy encode the quantized data. Examples are JPEG, JPEG2000, H.261, H.263, and MPEG. In video coding, *motion compensation* is a common data reduction technique. Since the redundancy between two subsequent frames F_i and F_{i+1} of a video sequence generally is prominent,

successive frames are searched for similar objects. The storage of the affine transformation, which maps an object in F_i onto F_{i+1} plus the encoded error of this prediction, is less demanding on the bit rate than the unpredicted frames. Motion compensation is incorporated into all MPEG coding standards.

Figure 4.2 demonstrates the idea of digital signal processing for a hybrid codec¹. On the left hand side, the *encoding* process is split into its component details: pre-processing, i.e., data gathering, transformation, quantization, and entropy encoding. On the right hand side, the *decoding* or *decompression* process is demonstrated. The difference between the original signal and the decoded signal is the content of the *error* signal, which reflects the distortion between input and output.

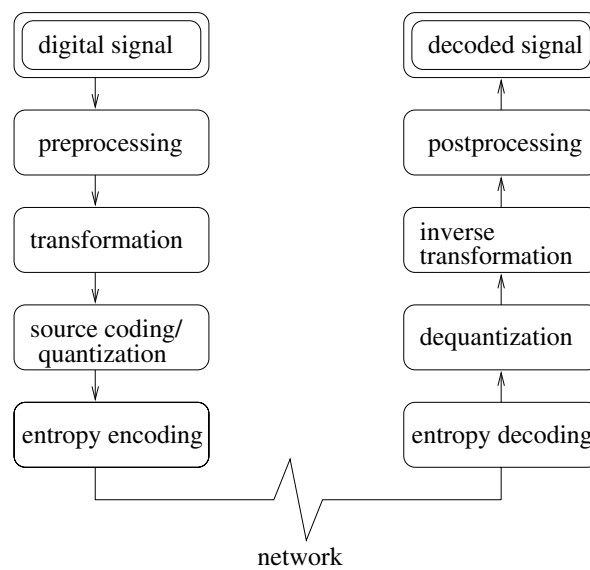


Figure 4.2: Hybrid coding for compression.

A second way to classify compression algorithms is according to the trade-off between a high compression rate², and the cost (i.e., time), and quality of compression. In general, the higher the compression rate of lossy algorithms is, the poorer is their perceived quality. Many multimedia applications further demand a low encoding and decoding variance.

The subdivision into *symmetric* and *asymmetric* approaches allows yet another classification of compression algorithms. It evaluates the coding costs at the encoder and the decoder. *Symmetric* techniques require approximately the same amount of time for both the encoding and the decoding process. Examples are the transformation-based cosine and wavelet transforms. *Asymmetric* techniques are very costly for the encoder, while the decoder is less complex [ES98]. An example of an asymmetric coding technique is fractal coding [BH93]. Asymmetric approaches are used in applications where

¹Codec = (en)coder / decoder

²The compression rate C_r is the relation between the amount of data of the original signal compared to the amount of data of the encoded signal:

$$C_r := \frac{\text{amount of data (original signal)}}{\text{amount of data (encoded signal)}}$$

the encoding is performed only once, and plenty of time is available, but the decoding is time-critical (e.g., a video server in the Internet).

Finally, compression techniques can be classified according to their *robustness*, i.e., according to the impact of transmission errors on the signal quality. Especially in real-time multimedia networks, the abdication of compression rate in favor of robustness can be desirable. Errors or data packets lost by the network should have a minor impact on the quality of the decoded signal.

Table 4.1 gives an overview of coding classifications.

entropy / source / hybrid coding
quality
compression rate
encoding / decoding cost
symmetry
robustness

Table 4.1: Classification of compression algorithms.

4.3 Nyquist Sampling Rate

What the human ear perceives as *sound* are physically small (analog) changes in air pressure that stimulate the eardrum. A digital system can handle only discrete signals, though. The conversion analog-discrete is realized through sampling. The level of the analog signal is measured at short intervals so that sound is represented by a sequence of discrete values.

If the sampling process of a signal f is represented as the multiplication of the continuous signal f by a sampling signal s at uniform intervals of T , which is an infinite train of impulse functions $\delta(kT)$, then the sampled signal f_s is [GR98]:

$$f_s(t) = f(t)s(t) = f(t) \left(\sum_{k=-\infty}^{\infty} \delta(t - kT) \right).$$

The impulse train s is a periodic function. If we take its Fourier transform, the sampled signal becomes

$$f_s(t) = f(t) \frac{1}{T} \sum_{p=-\infty}^{\infty} e^{ip\omega_0 t},$$

with the constant $\omega_0 = \frac{2\pi}{T}$. The spectrum $\hat{f}_s(\omega)$ of the sampled signal can be determined by taking the Fourier transform of $f_s(t)$, which results in [GR98]:

$$\hat{f}_s(\omega) = \frac{1}{T} \sum_{p=-\infty}^{\infty} \hat{f}(\omega - p\omega_0). \quad (4.1)$$

Equation (4.1) states that the spectrum of the sampled signal is the sum of the spectra of the continuous signal repeated periodically at intervals of $\omega_0 = \frac{2\pi}{T}$. Thus, the continuous signal f can be perfectly recovered from the sampled signal f_s under the condition that the sampling interval T is chosen such that

$$\frac{2\pi}{T} > 2\omega_B \quad \Leftrightarrow \quad T < \frac{\pi}{\omega_B}, \quad (4.2)$$

where ω_B is the bandwidth of the continuous signal f . If condition (4.2) holds for the sampling interval T , then the original signal can be perfectly reconstructed using a low-pass filter with bandwidth ω_B .

The above assumptions of a band-limited input signal and an *ideal* low-pass filter for the analog-digital and digital-analog conversions are not always encountered in this rigidity in applications. Nevertheless, this theory, commonly known as the *Nyquist sampling rate*, means that in principle, a digital audio system can be designed which contains all the perceptible information of its analog counterpart. It forms the base for the whole framework of digital audio processing (see Chapter 5).

Chapter 5

Digital Audio Denoising

It would be possible to describe everything scientifically, but it would make no sense; it would be without meaning, as if you described a Beethoven symphony as a variation of wave pressure.

– Albert Einstein

5.1 Introduction

When audio was still stored on analog media such as magnetic tape, duplication was inevitably accompanied by deteriorated quality. Random additive background noise is a type of degradation common to all analog storage and recording systems. The introduction of high-quality digital audio media, e.g., CD-ROM, *digital audio tape* (DAT), or *digital versatile disc* (DVD) has raised general expectations with regard to sound quality for all types of recordings. This has increased the demand for the restoration of qualitatively degraded recordings.

Noise in audio signals is generally perceived as *hiss* by the listener. It is composed of electrical circuit noise, irregularities in the storage medium, and ambient noise from the recording environment. The combined effect of these sources is generally treated as one single noise process. Many years of research have been devoted to audio denoising. From the early beginnings, mathematical transforms have provided the fundamental base for this demanding task. Random noise has significant components at all audio frequencies, thus simple filtering and equalization procedures are inadequate for restoration purposes [GR98]. The classic least-squares work of Wiener [Wie49] placed noise reduction on a firm analytic footing and still forms the basis of many noise reduction methods. In the field of speech processing, particularly in the domain of telephony, a large number of techniques have been developed for noise reduction (see e.g. [LO79] and [Lim83]), and many of these are more generally applicable to general noisy audio signals.

In Section 5.2, we consider some standard audio denoising approaches that are appropriate for general audio signals. For an exhaustive discussion, the reader is referred to the very readable book by Godsill and Rayner [GR98]. Based on the theoretical discussions in the PhD Thesis of Jansen [Jan00]

on wavelet thresholding and noise reduction, Section 5.3 provides the theory of wavelet-based audio denoising. Our own contribution to this chapter is presented in Section 5.4, where we suggest the implementation of a wavelet-based audio denoiser that allows flexible parameter settings and is suitable to teach the wavelet-based approach to students of engineering and computer science.

5.2 Standard Denoising Techniques

Noise in an audio signal denotes a perturbing, generally unwanted signal in some or all frequency bands. With the notion of noise, we enter a process that is not perfectly deterministic. In digital systems, it is the instantaneous voltage of noise which is of interest, since it is a form of interference that alters the state of the signal. Unfortunately, instantaneous voltage is not to be predictable. Therefore, noise can only be quantified statistically. [Wat95]

As mentioned above, noise models are based on the underlying assumption that a pure, undisturbed signal f is corrupted by noise η to result in the actual observed signal y :

$$y = f + \eta, \quad (5.1)$$

where y , f , and η are vectors of N samples. As η is a probabilistic variable, y is non-deterministic and only the ‘clean’ signal f is perfectly known. If we assume that the expectation $\mathbf{E}\eta$ of η is zero, then the covariance matrix C_η of η is

$$\mathbf{E}[(\eta - \mathbf{E}\eta)(\eta - \mathbf{E}\eta)^T] = \mathbf{E}\eta\eta^T =: C_\eta.$$

The matrix entries on the diagonal denote the variances $\sigma_i^2 = \mathbf{E}\eta_i^2$. If the covariance matrix C_η is a diagonal matrix, in other words: $\mathbf{E}\eta_i\eta_j = 0$ for $i \neq j$, then the noise is called *uncorrelated* or *white noise*. If all data points are deduced from the same probability density, the noise is said to be *identically distributed* [Jan00]. This implies:

$$\sigma_i^2 = \sigma^2 \quad \text{for } i = 1, \dots, N.$$

An important density model is the joint Gaussian:

$$\phi(C_\eta) = \frac{1}{(2\pi)^{N/2} \sqrt{\det C_\eta}} e^{-\frac{1}{2}\eta^T C_\eta^{-1} \eta}.$$

If Gaussian noise variables are uncorrelated, they are also independent. A classical assumption in regression theory is the assumption of independent, identically distributed noise. [Jan00]

Many approaches have been researched for the restoration of a degraded audio signal. Obviously, an ideal system processes only those samples that are degraded, leaving the undistorted samples unchanged. A successful noise restoration system thus encompasses two tasks [GR98]:

- *Detection.* The detection procedure will estimate the value of the noise η , in other words it decides which samples are corrupted.
- *Reconstruction.* An estimation procedure attempts to reconstruct the underlying *original* audio data.

Criteria for the successful detection of noise include minimum probability of error and related concepts. Strictly speaking, the goal of every audio restoration scheme is to remove only those artifacts which are audible to the listener. Any further processing increases the chance of distorting the perceived signal quality, while being unnecessary on the other hand. Consequently, the determination of the best value in the trade-off between the audibility of artifacts and the perceived distortion as a result of the processing would require the consideration of complex psychoacoustic effects in the human ear [Gol89] [ZGHG99] [vC93]. Since such an approach is difficult both to formulate and to implement, we restrict our considerations here to criteria that are mathematically understood.

5.2.1 Noise Detection

Simple but very effective noise detection methods are based on the assumption that most audio signals contain little information at high frequencies, while the noise model described above has spectral content at all frequencies. A high-pass filter helps to enhance these high-frequency components of the noise relative to the signal, which can then easily be detected by thresholding the filtered output. This principle is the basis of many kinds of analog denoising equipment, as well as of digital tools. In [GR98], a number of autoregressive denoising techniques are proposed that are based upon prior knowledge of the signal and the noise.

5.2.2 Noise Removal

Traditionally, methods for noise reduction in audio signals are based on short-time processing in the spectral (i.e., frequency) domain. The reason for the success of these methods is that audio signals are usually composed of a number of line spectral components. Though they are time-varying, they can be considered as fixed over a short analysis time (generally of about 0.02 seconds). Thus, the analysis of short windows of data in the frequency domain concentrates the energy of the signal into relatively few frequency ‘bins’ with a high signal-to-noise ratio [GR98]. Processing then is performed in the frequency domain, often based on the short-time Fourier transform of the noisy signal y , i.e., based on $\tilde{S}y$ (see Section 1.4.2) in order to estimate the spectrum of the ‘clean’ data f :

$$\widehat{f_{\text{est}}} = \nu(\tilde{S}y), \quad (5.2)$$

where ν is a function that performs noise reduction on the spectral components. The estimated spectrum $\widehat{f_{\text{est}}}$ is processed with the inverse short-time Fourier transform to obtain a time-domain signal estimate f_{est} of the ‘clean’ signal f . Many possible variants have been proposed for the processing function ν , which might be used to perform noise reduction in the frequency domain, some based on heuristic ideas, and others based on a more rigorous foundation such as the Wiener, or maximum

likelihood estimation. In the scope of this work, we do not detail the function ν . See [GR98] for more details.

Most methods based on the assumption of Equation (5.2) lead to a significant reduction of background noise in audio recordings. However, there are some drawbacks which inhibit the practical application of these techniques without further modification. The main one being the residual noise artifact often referred to as *musical noise*. It arises from the randomness inherent in the crude estimate of the signal power spectrum. Methods to improve spectral domain denoising based on the short-time Fourier transform encompass the approach to make a more statistically stable estimate \widehat{f}_{est} based on different *time samples* of the signal, i.e., varying estimates over time. Another way to improve the quality of restoration is to devise alternative noise suppression rules based upon sophisticated criteria. The literature proposed a number of techniques for achieving either or both of these objectives. They encompass the elimination of musical noise by over-estimation of the noise in the power spectrum, maximum likelihood, and minimum mean-squared error estimators.

All the above discussion is based on the assumption of a Fourier transform-based implementation of noise reduction methods. Recent work has seen noise reduction performed in alternative basis expansions, in particular the wavelet domain [Mon91] [Jan00]. Due to their multiresolution property (see Section 1.6), wavelets have useful localization properties for singularities in signals [MH92]. Hence, wavelet transform-based denoising techniques promise to overcome most of the above inconveniences.

5.3 Noise Reduction with Wavelets

This section reviews the theory of wavelet-based audio denoising before we present our own implementation and our empirical results in Section 5.4. The mathematical discussion is based on the notation introduced above.

5.3.1 Wavelet Transform of a Noisy Audio Signal

In Observation 1.1 we have seen that the wavelet transform is a linear mapping. If the noisy signal y of Equation (5.1) is wavelet-transformed with regard to the wavelet ψ , it results in

$$\tilde{y}_\psi = \tilde{f}_\psi + \tilde{\eta}_\psi, \quad (5.3)$$

i.e., the model of our noisy signal remains unchanged. The covariance matrix of the noise in the wavelet domain is thus

$$\mathbf{E}\tilde{\eta}_\psi\tilde{\eta}_\psi^T = \tilde{C}_{\eta,\psi}, \quad (5.4)$$

where $\tilde{C}_{\eta,\psi}$ denotes the wavelet transform of the covariance matrix C_η with regard to the wavelet ψ with nonstandard decomposition [Jan00]. Equation (5.4) holds for a general linear transform.

5.3.2 Orthogonal Wavelet Transform and Thresholding

If the wavelet transform is orthogonal and $C_\eta = \sigma^2 \mathbf{I}$, then $\mathbf{E} \tilde{\eta}_\psi \tilde{\eta}_\psi^T = \sigma^2 \mathbf{I}$. This means [Jan00]:

Observation 5.1 *An orthogonal wavelet transform of identically distributed white noise is identically distributed and white.*

In practical regards, Observation 5.1 means that an orthogonal wavelet transform decorrelates a signal with correlations. On the other hand, uncorrelated noise remains uncorrelated.

It is a general observation that statistically, the absolute amount of noise is identical for all coefficients in the time–scale domain. This means that *small* absolute wavelet–transformed coefficients are dominated by the noise, while *large* absolute coefficients contain mostly signal information, and only a minor amount of noise information. A noisy signal might thus be *denoised* by analyzing the wavelet–transformed coefficients and eliminating the small coefficients of the time–scale domain, thus intensifying the impact of the large values. More precisely, a wavelet–based audio denoising algorithm is based on three assumptions:

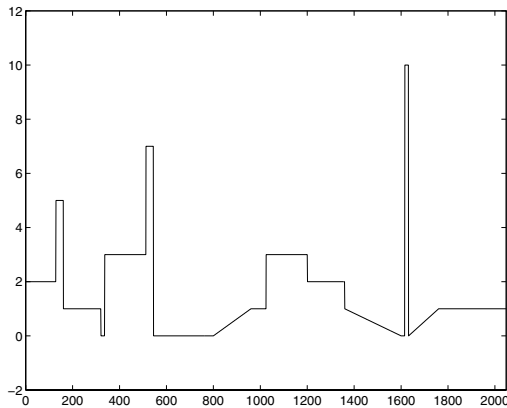
- the absolute amount of noise is spread equally over all coefficients,
- harmonic content like music or speech is highly correlated and thus produces larger coefficients than noise, which is highly uncorrelated,
- the noise level is not *too* high: We can recognize the signal and the signal’s coefficients in the time–scale domain.

The removal of the small coefficients thus constitutes noise removal. Wavelet thresholding combines simplicity and efficiency and is therefore a promising noise reduction method. It was first introduced by Donoho and Johnstone [Don93a] [Don93b] [DJ94] [Don95], and expanded quickly [CD95] [CYV97].

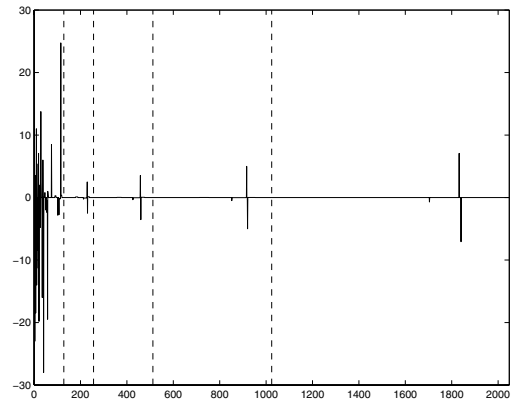
Figures 5.1 (a) and (b) demonstrate a ‘clean’ audio signal f and its coefficients \tilde{f}_ψ in the time–scale domain after wavelet–transformation with regard to the Haar filter bank. Figures 5.1 (c) and (d) show the noisy audio signal y and its wavelet–transformed counterpart \tilde{y}_ψ . As can be clearly seen, the uncorrelated noise in the time domain is reflected in the time–frequency domain by many small coefficients in all frequency bands. Figure 5.1 (e) shows the denoised signal f_{est} in the time domain after thresholding the time–scale domain with a threshold set to $\lambda = 1$. The estimate is not identical to the original signal, though.

There exist miscellaneous wavelet–based audio denoising techniques which differ in their treatment of the large absolute coefficients in the time–scale domain, and have different implications for the subjective perception of a denoised audio signal. These include:

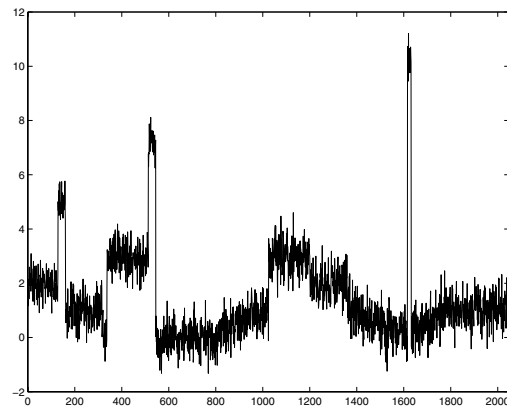
- *Hard thresholding.* When the coefficients on some or all scales that are below a certain threshold are set to zero while the coefficients superior to the selected threshold remain unaltered, one speaks of hard thresholding (see Figure 5.2 (a)). This is the ‘keep–or–kill’ procedure. Given an increasing threshold, this policy exhibits subjectively disturbing artifacts.



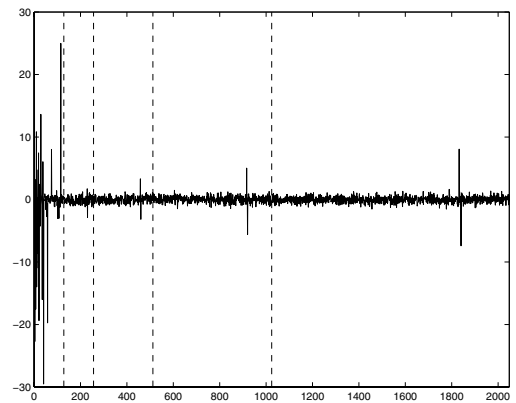
(a) Original 'clean' audio signal in the time domain.



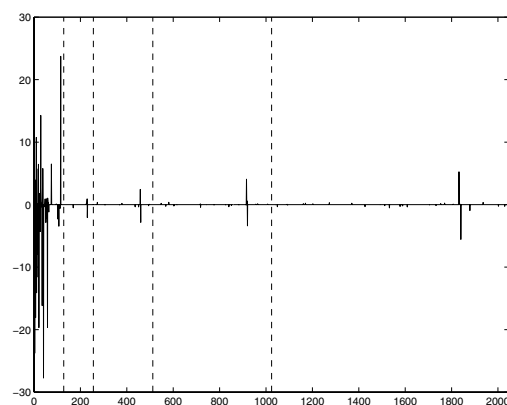
(b) Wavelet-transformed original 'clean' signal.



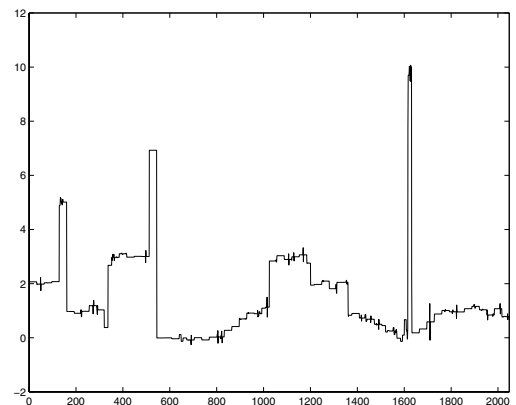
(c) Noisy audio signal in the time domain.



(d) Wavelet-transformed noisy audio signal.



(e) Wavelet-transformed noisy audio signal after soft thresholding with $\lambda = 1$.



(f) Denoised audio signal.

Figure 5.1: Effect of wavelet-based thresholding of a noisy audio signal. (a) and (b): Original 'clean' signal and its wavelet transform with the Haar filter bank. (c) and (d): Audio signal with identically distributed white noise. The noise is spread equally over all coefficients of the time-scale domain, and the signal's singularities are still distinguishable. (e) and (f): Coefficients in the time-scale domain after soft thresholding with $\lambda = 1$ and the resulting denoised audio signal. (Courtesy of [Jan00])

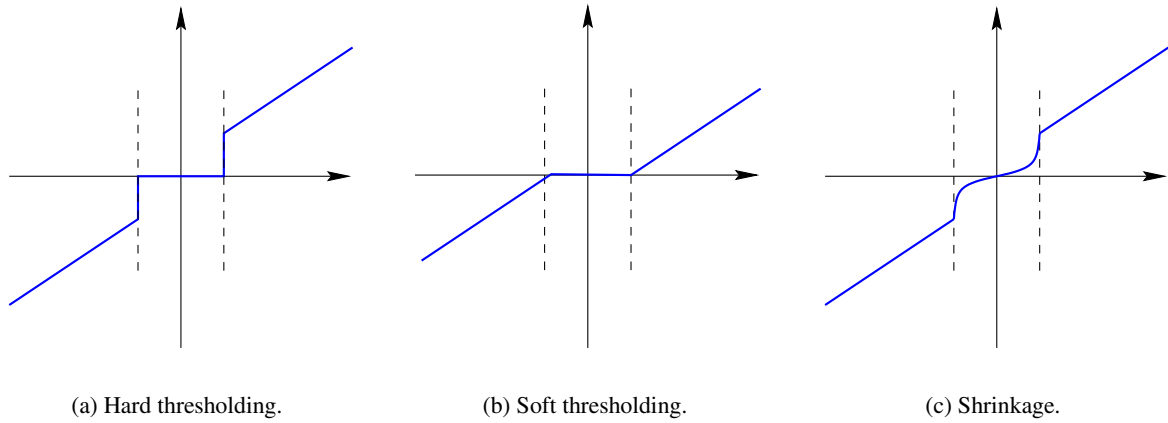


Figure 5.2: Hard and soft thresholding, and shrinkage.

- *Soft thresholding.* When the coefficients on some or all scales below a threshold are set to zero, and additionally, all coefficients above the selected threshold are shrunk by the value of the threshold, one speaks of soft thresholding (see Figure 5.2 (b)). In so doing, this procedure attenuates the range of the wavelet coefficients and smoothes the signal, thus modifying its energy [Roa96].
- *Shrinkage.* A compromise between the above two thresholding policies is presented in Figure 5.2 (c). It preserves the highest coefficients but has a smooth transition between the cut-off and the maintained coefficients. Several shrinkage functions and techniques have been proposed by the team around Gao [GB97] [SPB⁺98] [Gao98]. Some of them depend on more than one threshold, others do not rely on thresholds at all. A shrinkage class derived from Bayesian modeling is discussed in [Jan00].

In general, the sophisticated shrinkage schemes are computationally very expensive. As soft thresholding is agreed to be a good compromise between computational complexity and performance [Jan00], we will concentrate on hard and soft thresholding in this work.

5.3.3 Nonorthogonal Wavelet Transform and Thresholding

If the assumptions on the wavelet transform are loosened and we allow biorthogonal wavelet transforms to analyze a signal with identically distributed white noise, Equation (5.4) tells us that the coefficients in the time–scale domain will be correlated and not stationary. In this case, we could apply scale–dependent thresholds, i.e., vary the level of the threshold with the scale under consideration [LGOB95].

However, there are still problems with scale–dependent thresholding [Jan00]:

- A number of noisy coefficients always survive the threshold.

- The error-minimizing process needs sufficiently many coefficients in each scale to work well; coarser resolution levels may lack this number and may not find a separate threshold for this level.

It is therefore common practice to return to one single threshold even for nonorthogonal wavelet transforms, but to apply another heuristic about the wavelet transform: If a coefficient in the time-scale domain is large (in absolute value) due to a signal's singularity, we may expect that the corresponding coefficients in the next coarser scale will also be large in absolute value. Thus, a signal's singularity impacts all scales from fine to coarse up to a certain level, and the signal's specific features strike a wide range of scales. Conversely to this characteristic of a signal, white noise is a local phenomenon: the singularity of a noise does not penetrate into the coarser scales.

The threshold method that we have implemented in our own wavelet-based audio denoiser (see Section 5.4) is the tree-structures thresholding [CDDD00] [Bar99]. A *tree* is a set of wavelet coefficients. For each element in this set, the coefficients in the time-scale domain at 'the same location', but also in the next coarser scale belong to the set. The name tree is derived from the fact that two different coefficients in a given fine scale share one single coefficient in the next coarser scale, hence resulting in a branched structure.

5.3.4 Determination of the Threshold

A central question in many threshold-based applications is how to determine a suitable threshold for a specific purpose. A threshold subdivides a set into two 'yes/no'-subsets. Hence, in the audio denoising context, a threshold selection is the trade-off between removal of noise and removal of too much *original* audio. A small threshold yields a result close to the input, and might still contain too much noise. A large threshold, however, results in many zero coefficients in the time-scale domain which might destroy some of the signal's singularities. In audio coding, this means that the signal is 'denuded' of its timbre, resulting in the negative audible artifacts of a thump signal.

In search of a good threshold selection, a very common approach is to minimize the mean square error. It requires the original signal for comparison with the noisy one. Since the undisturbed signal is normally not known, an optimal threshold for minimization of the mean square error is seldom found. Estimation of the minimum has been investigated in many articles [DJ95] [JB99] [Nas96].

With our wavelet-based audio denoiser, we show that the (one-dimensional) wavelet transform is fast enough to allow real-time application. The audio denoiser does not implement a fixed threshold, but it is designed to allow a flexible threshold to be set by the user.

5.4 Implementation of a Wavelet-based Audio Denoiser

Our contribution to the use of the wavelet transform in digital audio applications is the implementation of a wavelet-based audio processor with the following features:

- proof of the efficiency of the (discrete, dyadic) wavelet transform for real-time applications,

- ‘proof by implementation’ of the wavelet-based denoising theory [Jan00] that we discussed in Section 5.3,
- subjective judgment of the chosen parameter settings by ‘hearing’ the wavelet filters, and
- ‘seeing’ the wavelets and the effect of multiresolution analysis.

Besides this, our implementation pursues a second goal, independent of this technical one: It is a didactic tool used in our multimedia course. This aspect will be elaborated in Part III.

The implementation has been carried out within the scope of a master’s thesis [Böm00], written in collaboration with the company I3 Srl. in Rome, Italy. Windows95/98, equipped with the Microsoft Visual Studio, was used as the development platform with the C++ programming language. The program is composed of 59 classes, subdivided into the five categories: wavelet transform, framework core classes, framework extensions, filters, and Windows GUI. It totals 10.400 lines of code. The audio tool itself was presented at a workshop on signal processing [SHB00], while the denoising feature, its theory, and empirical evaluation were presented at [SHB01].

5.4.1 Framework

A framework for digital audio processing has been developed whose core classes provide the interfaces and the implemented classes. For details on the class structure of the implementation, we refer to [Böm00]. All sources, destinations, and modifiers of the data flow are implemented as independent extensions. This allows us to extend the audio framework by new features.

A graphical user interface (GUI) has been developed for the Windows platform (see Figure 5.3). It uses a simple chain with two readers (i.e., soundcard or file input) and two writers (i.e., soundcard or file output). Any number of implemented features — which we will call *filters* in this context — can be arranged in-between. The flow of audio data is symbolized by arrows. Select buttons on the user interface are used to activate the sources and destinations.

In Figure 5.3, the setting is as follows: The audio data are read directly from the soundcard, e.g., they are captured by a microphone. The processed data are written to the soundcard output. Concerning the digital processing, a difference listener is started prior to adding a noise generator. The forward wavelet transform paves the way for a wavelet denoiser. The display of the coefficients in the time-scale domain is added before the wavelet transform is reversed and the difference listener ends.

The choice and the order of the filters is very flexible. When the user wants to add an implemented filter, the dialog box in Figure 5.4 (a) opens. For each selected functionality, a window pops up either to allow further parameter setting (see Figure 5.4 (b): the parameters of the noise generator are set) or to display a feature (see Figure 5.4 (c): the coefficients in the time-scale domain are visualized). The filters are applied to the input audio stream in the order of their arrangement in the GUI (see Figure 5.3), from top to bottom. The order of the selected filters can be adjusted by moving them up or down (see Figure 5.3: the *wavelet display* is marked). The results of all actions are sent directly to the soundcard, respectively, to the output file.

The presented setup allows to efficiently find good parameter settings for the denoiser. For the forward and inverse wavelet transforms, all standard wavelet filters are implemented: Daubechies, Coiflets,

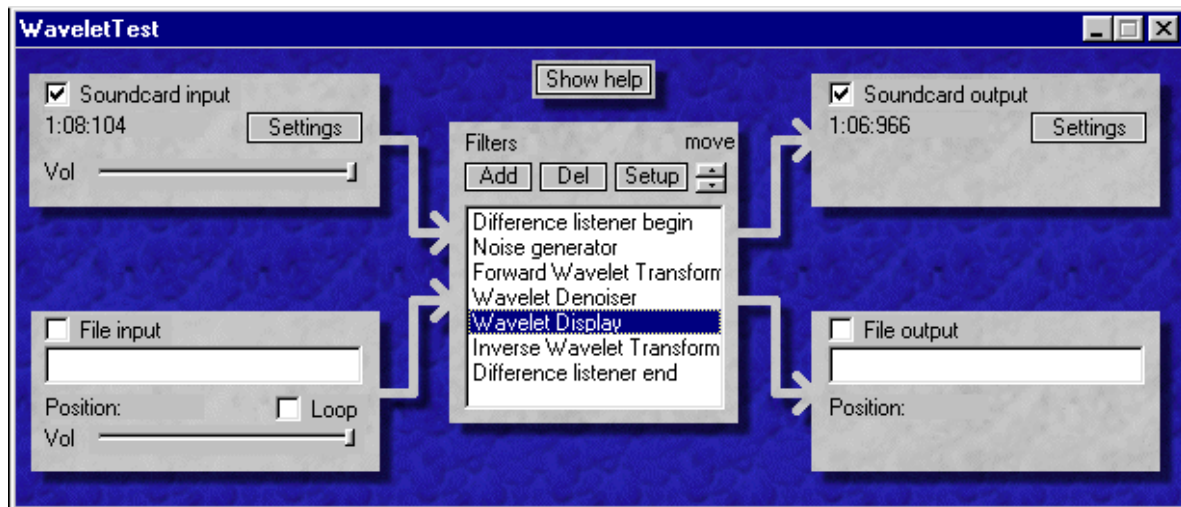


Figure 5.3: Graphical user interface of the wavelet-based audio tool. Audio data can be read from soundcard/file and are written to soundcard/file. In-between, a number of filters are applied.

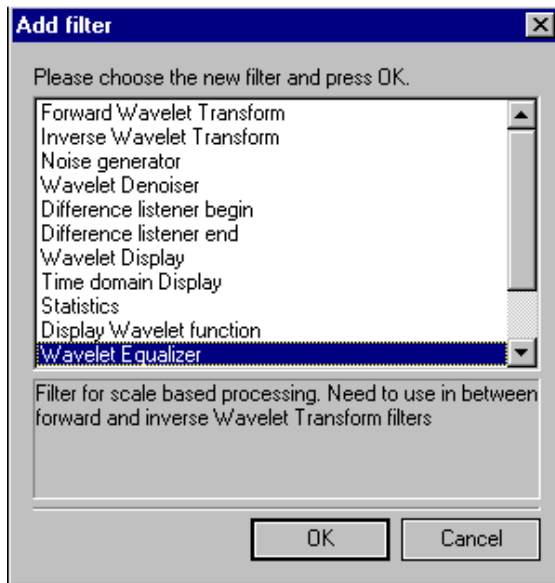
Symlets, Biorthogonal, Battle–Lemarié, Spline. For the definition of these filters see [Dau92]. The choice of how to handle the boundary problem for the audio signal is also left to the user: he/she can select from zero padding, mirror padding, and circular convolution (see Section 3.3).

Two sample functionalities that have proven to be very useful in the teaching/learning aspect (see Part III) are:

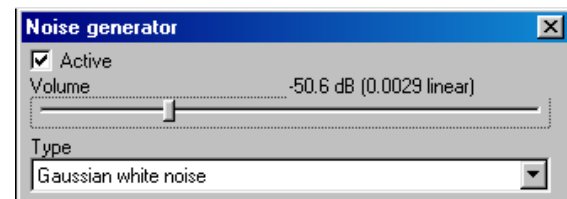
- *Display of the multiscale analysis.* A window displays all wavelet coefficients in time versus scale. Here, the amplitude of a coefficient is represented by colors. Every modification of the wavelet coefficients can be monitored directly. In our example in Figure 5.3, the effect of different denoising parameters can be followed visually (see Figure 5.4 (c)).
- *Visualization of the chosen wavelet function ψ* for the wavelet transform. This is achieved by means of a simple trick: Setting to zero all coefficients in the time–scale domain except one single scale results in the time–domain wavelet function created by the inverse transform. Figure 5.5 (a) shows the filter that allows control of the scale, temporal position, and amplitude of the single non–zero coefficient. Adding a time domain display filter after the inverse wavelet transform permits the wavelet function to be seen and explored. Changing scale and time shows the effects of dilation and translation of a wavelet (see Figure 5.5 (b)).

5.4.2 Noise Reduction

This section concentrates on the noise reduction by our audio tool. In order to study noise, the audio signal that is read by our tool can be disturbed: A noise generator adds white noise to the input audio file. A parameter determines whether the noise distribution is uniform or Gaussian, and the amount of noise can be adjusted between $-\infty$ dB (no noise) and 0 dB (maximum noise energy).



(a) Add filter dialog box.

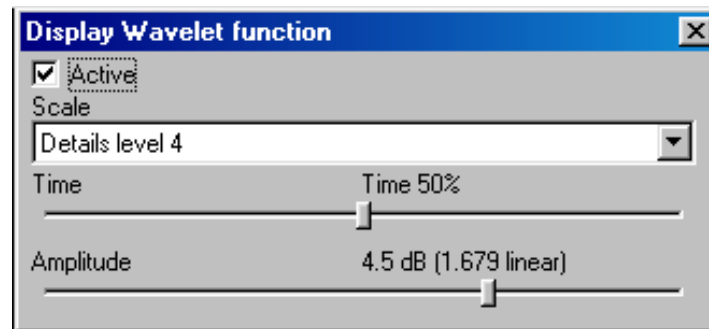


(b) The noise generator produces white noise.

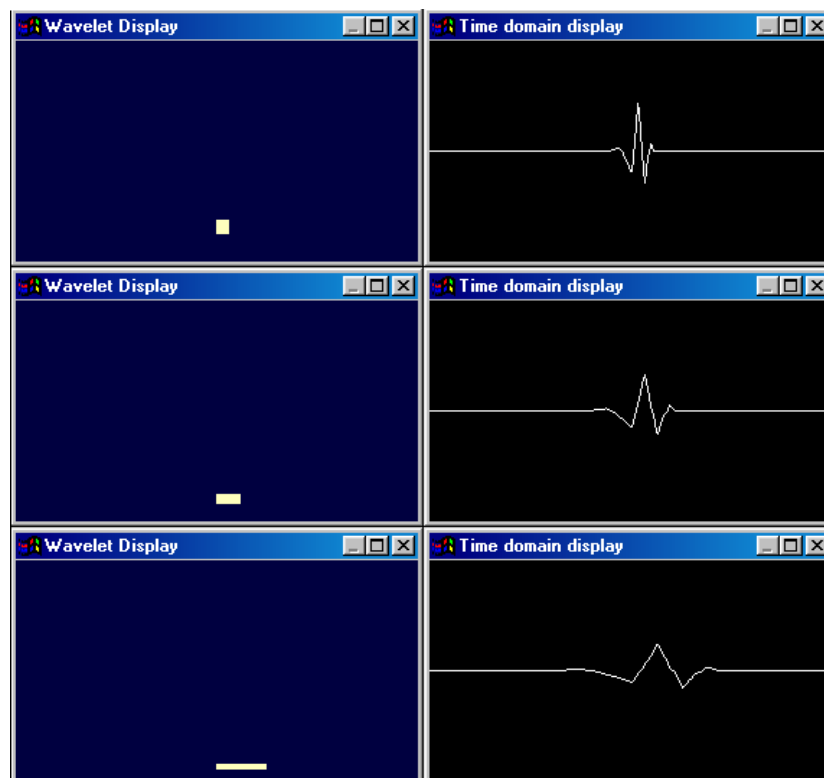


(c) Visualization of the time-scale domain. The higher the wavelet coefficient in a scale is, the darker the area.

Figure 5.4: Selected features of the wavelet-based digital audio processor.



(a) Filter to set all but one wavelet scale parameters to zero.



(b) With the dialog box shown in (a), time-scale coefficients of scales 4, 5, and 6 have been selected accordingly. The inverse wavelet transform thus results in the display of the underlying wavelet function at the different scales.

Figure 5.5: Visualizations of the time-scale domain and of the time domain. When all but one of the coefficients in the time-scale domain are set to zero, the inverse wavelet transform results in the display of the wavelet function.

In our example in Figure 5.3, the entire set of actions is surrounded by the difference listener. This allows us to concentrate on the difference between original and denoised signal. Complete silence indicates perfect reconstruction: The noise that has been added to the signal has been perfectly removed, and the signal itself has not been modified. Music in the difference listener, however, corresponds to side-effects of the denoising process. A slightly different setup, with the *difference listener begin* applied *after* the noise generator, would allow one to hear the noise that has been removed by the denoiser.

The wavelet denoiser (see Figure 5.6 (a)) is applied before the inverse wavelet transform synthesizes the audio data. The wavelet denoiser realizes hard or soft thresholding (see Section 5.3.2). The parameters of the wavelet denoiser include the type of algorithm, the cut-off threshold, the number of levels to be treated (see Section 5.3.3), and whether the padding coefficients of the boundaries are to be included for thresholding. Thus, the parameters allow flexible control of the denoising process.

In addition to the audible output, the presented audio tool can also visualize the performed denoising: The application of a time domain display for the noisy signal and a second time domain display after removal of the noise is demonstrated in Figure 5.6 (b). This service is especially suited to demonstrate the denoiser when only visual media are available (as in this book).

5.4.3 Empirical Evaluation

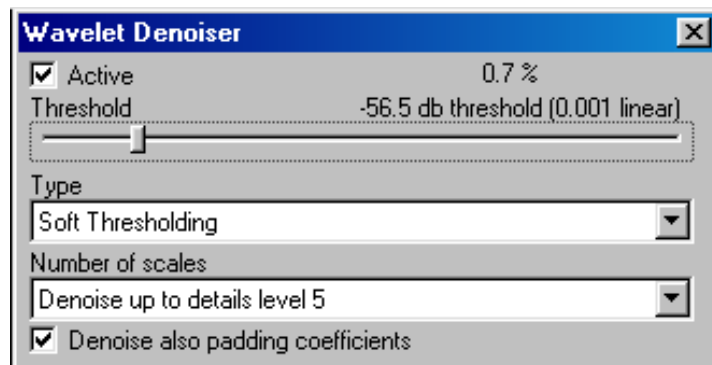
From the didactic point of view of teaching our students, our wavelet denoiser aims to provide practical experience with the wavelet-based denoising concepts presented in Section 5.3.

- Can we perceive the difference between uniform and Gaussian white noise in audio?
- What is the audible difference between hard and soft thresholding?
- How much noise can be added to a signal without irreversible deterioration?
- What is the effect of the different padding schemes for coefficients?

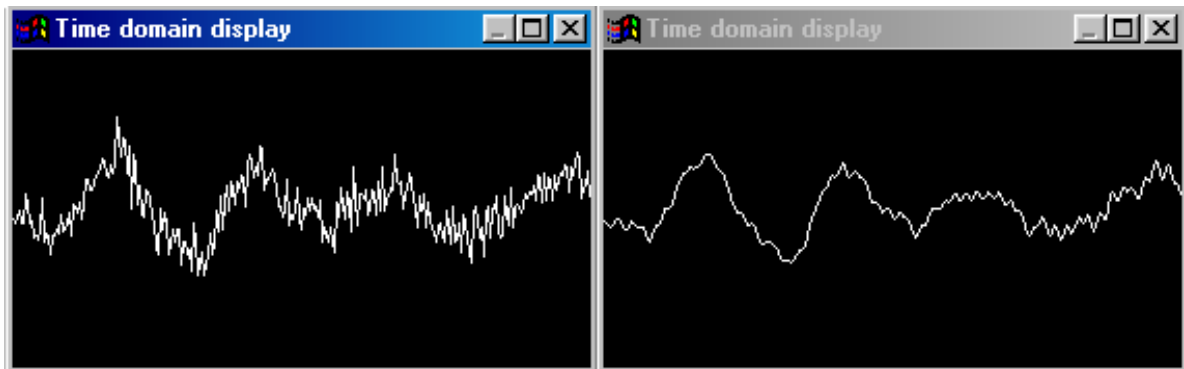
As human perception is still not totally understood and models vary strongly, directly *hearing* the result of a parameter setting – and judging it instantly – is still the easiest and most reliable way to get an answer to these questions. This section presents some of our empirical evaluation results.

As we have stated in Section 5.3.4, the setting of an appropriate threshold for denoising audio data always represents a trade-off between the removal of noise and the removal of genuine audio data. We present the quantitative error estimate for audio data erroneously removed during the denoising process. The error estimate requires the availability of the undistorted data in comparison to the noisy signal, a condition that was met for our purpose. Since the original signal f is known, a measure of the amount of error present in the cleaned data f_{est} is obtained by taking the root mean square deviation of the cleaned signal from the pure signal as follows [RKK⁺99]:

$$\bar{E} = \sqrt{\frac{1}{N} \sum_{i=1}^N (f_{\text{est}}(t_i) - f(t_i))^2}, \quad (5.5)$$



(a) The wavelet denoiser can perform hard or soft thresholding.



(b) Display of the time domain before and after application of the wavelet denoiser.

Figure 5.6: Visible results of the denoising process.

where N is the length of the time series. A similar quantity to Equation (5.5) is calculated for the noisy signal y and denoted by \bar{E} . The error estimate is then given by \bar{E}/E and the following statements hold:

- The relation $\bar{E}/E < 1$ stands for successful noise removal, whereas
- the relation $\bar{E}/E \geq 1$ means that data of the original signal has been removed together with the noise.

Table 5.1 shows the evaluation results for the music file `dnbloop.wav`, which has a wide frequency usage and some short transients. The wavelet filter bank was a Battle–Lemarié filter with 49 taps, the boundary policy was set to mirror padding, the denoiser was set to soft thresholding using the first five levels of wavelet coefficients and including the padded coefficients.

	Objective assessment		Subjective assessment	
Noise	Threshold	\bar{E}/E	Threshold	\bar{E}/E
-37 dB	-58 dB	0.956	-50 dB	1.121
-34 dB	-50 dB	0.977	-47 dB	1.063
-32 dB	-50 dB	0.921	-45 dB	1.012
-30 dB	-51.5 dB	0.896	-43.5 dB	0.940
-27 dB	-44.5 dB	0.840	-40 dB	0.871

Table 5.1: Evaluation of the wavelet denoiser for `dnbloop.wav`. The noise amplitude is given in dB. Objective assessment yields the minimum error estimate \bar{E}/E . Subjective threshold setting is not optimal, but approaches the minimum with increasing noise.

For the objective assessment, the threshold was set to yield a minimum error [Böm00]. For a fixed E , this turned out to be the minimum \bar{E} . The subjective assessment revealed the average rating of five probands, where the least noticeable noise in the setting with the difference listener (see Section 5.4.2) was indicated.

The error estimate in Table 5.1 reveals that increasing noise also requires an increasing threshold parameter for the denoiser. Furthermore, the subjectively adjusted threshold is in all cases much higher than the automatically chosen threshold. As the objective assessment was constructed to result in minimum error, the subjective setting by the ear cannot deliver better results. The minimum error thresholds all result in an error estimate below 1, the algorithm thus has proven its success. The results of the subjective threshold adjustment can be interpreted as follows: The less noise that is added, the more difficult it is for people to detect it at all. A denoising threshold where the parameter is set too high might then result in erroneous removal of audio data, but this will still be below audible stimuli.

As the choice of threshold parameters is crucial, we have used the real-time aspect of our tool to compare subjective parameter settings to automatic minimum error settings. With low noise, human perception does not or nearly does not hear the artifacts introduced by the denoising algorithms. The higher the distortion gets (i.e., with increasing noise level), the better the perceived nuisance, and the better the ideal threshold can be approximated. This result is reflected by the error estimate of the two test series, where, with increasing noise, subjective assessment approaches objective assessment.

In the following chapter, we address the use of wavelet coding for still images for the purpose of compression.

Chapter 6

Still Images

Reducing a liter of orange juice to a few grams of concentrated powder is what lossy compression is about.

– Stéphane Mallat

6.1 Introduction

From the theoretical point of view, still image coding is a simple extension of one-dimensional coding techniques into the second dimension. Like audio coding, digital image coding has two major focal points of *recognition*, i.e., content-related processing, and *compression*. However, two aspects make up all the difference between audio and image coding:

- The number of sampling points in an image (i.e., the pixels) is generally far lower than the sampling points in an audio piece. Therefore, the boundary treatment in image coding becomes far more important.
- The human perception uses different ‘receptors’ to process the stimuli: while audio is captured with the ear, images are captured with the eye. Consequently, the perception of both varies largely.

In this chapter, we present novel applications of the wavelet transform on still images. Section 6.2 is oriented towards image *recognition* and demonstrates the application of multiscale analysis of the wavelet transform for boundary recognition. We make use of this feature to set up and implement an algorithm for semiautomatic object segmentation which has been presented in [HSE00]. In Section 6.3, we turn the focus on image *compression*. One strength of the wavelet transform is the flexibility in the selection of the parameters for a given coding problem. This strength is inconvenient since the freedom of choice also allows ineligible parameters to be set. In this second focal point, we present empirical parameter evaluations for image coding with regard to the implemented boundary policy [Sch02], the wavelet filter bank [SKE01b], and different decomposition strategies [Sch01b].

These evaluations help us to provide parameter recommendations for image coding problems. Section 6.4 generalizes a specific feature of JPEG2000 [SCE00a] [ITU00]: *regions-of-interest* (ROI) coding. A regions-of-interest is a region of the image (e.g., the face of a portrait) which is of specific importance to the observer. In scenarios where the image is compressed and/or distributed via the Internet, so that the quality of the image deteriorates at least temporarily, the ROI should be coded with maximum quality, with degraded quality only outside the ROI.

6.2 Wavelet-based Semiautomatic Segmentation

Image segmentation is an essential process in most subsequent image analysis tasks. In particular, many of the existing techniques for object-based image compression highly depend on segmentation results since compression techniques enjoy an enormous gain in performance once the *important* areas are known (cf. the discussion of the regions-of-interest in Section 6.4). Furthermore, object *recognition* is a major current research area. It aims to use the semantics of an image to retrieve specific features (e.g., all images containing a *Concorde* aircraft) from large data bases. Image segmentation, however, is a necessary pre-requisite for these tasks, and the quality of the segmentation affects the quality and reliability of the subsequent algorithms.

6.2.1 Fundamentals

Many techniques have been proposed to deal with the image segmentation problem. They are categorized as follows [HEMK98]:

- *Histogram-based techniques.* The image is assumed to be composed of a number of constant-intensity objects in a well-separated background. The segmentation problem is reformulated as one of parameter estimation followed by pixel classification [HS85].
- *Edge-based techniques.* The image edges are detected and then grouped into contours that represent the boundaries of image objects. Most techniques use a differentiation filter in order to approximate the first- or second-order image gradient. Candidate edges are extracted by thresholding the gradient magnitude [MH80] (see also Section 9.2).
- *Region-based techniques.* The goal is the detection of regions (i.e., connected sets of pixels) that satisfy certain predefined homogeneity criteria [Jai89]. In region-growing techniques, the input image is first tessellated into a set of homogeneous primitive regions. Then, similar neighboring regions are merged according to a certain decision rule (bottom-up). In splitting techniques, inhomogeneous regions of the initial entire image are subsequently divided into four rectangular segments until all segments are homogeneous (top-down).
- *Markov random field-based techniques.* The image is assumed to be a realization of a Markov random field with a distribution that captures the spatial context of the scene. The segmentation problem is then formulated as an optimization problem [DJ89].
- *Hybrid techniques.* The aim here is to offer an improved solution to the segmentation by combining the previous algorithms [HEMK98].

The search for automatic and reliable computer-based segmentation algorithms yet encounters two major problems:

- *Definition.* What is an object? The definition and detection of objects in still images is highly related to context. For example, in the image *Ostrich* presented in Figure 1.5, an object could be
 - the ostrich itself (in contrast to the background),
 - the neck of the ostrich (in contrast to its feathers),
 - the eye (in contrast to the neck),
 - the beak (in contrast to the eye), etc.
- *Detection.* The human eye works in a context-sensitive way. Looking at Figure 6.1, the human eye segments five horses. Since there are primarily only two different shades in the image (white and brown, respectively, gray in the printed version), no automatic detection system to date is capable of segmenting, e.g. the foal on the right hand side. As opposed to a computer, a human being has context-sensitive background information about ‘what a horse normally looks like’ and achieves the goal.

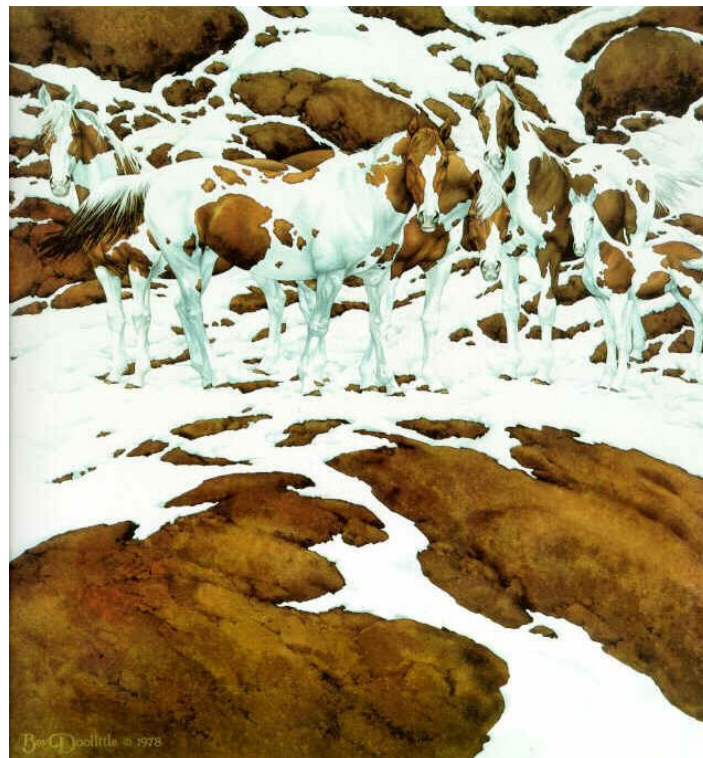


Figure 6.1: *Pintos* by Bev Doolittle, 1979. The human eye segments five horses. Until today, every automatic segmentation algorithm fails.

A major challenge for object segmentation is that most real-world objects have a highly complex structure. In automated processes, segmentation problems might occur for the following reasons.

(1) the background color is not uniform, (2) the object boundary color is not uniform, (3) a color that defines the object boundary at location (x_0, y_0) also exists in the foreground (i.e., in the interior of the object) and in the background, (4) the boundary is ‘fuzzy’, (5) parts of the object are occluded by other objects, and (6) the object of interest is not connected but is composed of more than one component.

Due to the above arguments, an intelligent automatic segmentation algorithm would require far more background information, knowledge, and ‘intelligence’ than today’s algorithm-based approaches. Concerning segmentation of objects in a general case, we claim that human interaction is still necessary. The human can define which object is of interest for his/her purpose. The algorithm then has the task to track this pre-defined object.

In the following section, we present a novel wavelet-based semiautomatic image segmentation algorithm which makes use of the multiresolution property of the wavelet transform and which is stable to the points 1–4 of the above list. The last two items, however, enter so deeply into psycho-visual science that we do not consider them in our approach.

6.2.2 A Wavelet-based Algorithm

The presented algorithm for still image segmentation was developed jointly with Thomas Haenselmann at our department [HSE00]. It is called *semiautomatic* since the user selects a piece of a boundary that separates the object from the background. The selected contour is then tracked. The boundary does not necessarily have to be sharp; e.g. the ‘lilty’ neck of the ostrich in Figure 1.5 presents a fuzzy transition. Thus, in contrast to other image segmentation algorithms, ours does not require a sharp boundary as the algorithm can follow any kind of visible transition between different regions.

The following steps compose the wavelet-based semiautomatic segmentation algorithm:

1. The user defines a convenient sample boundary by either selecting a starting and an ending point, both located on the object boundary, or by tracking a piece of boundary. In both cases, the algorithm constructs a sample rectangle of size $n_0 \times m$ pixels, where m is the distance between the selected points or the length of the trace. The sample rectangle is constructed such that the two selected points, respectively, the boundaries of the trace lay on the left and right edges of the rectangle at height $n_0/2$, where n_0 is set arbitrarily, but fixed. For the dyadic fast wavelet transform it is convenient to set $n_0 = 2^n$, so that we have implemented the algorithm with $n_0 = 128$. The so-defined sample rectangle provides information about ‘what the boundary between object and background looks like’.
2. The sample rectangle is rotated until it is parallel to the axes of a two-dimensional Euclidean coordinate system. Hence, the m columns which compose this sample rectangle all contain information about the transition from the object to the background.
3. The columns of the sample rectangle are wavelet-transformed as one-dimensional signals. Due to the construction of the algorithms, it is known that the object boundary is located somewhere between the first and the last pixel of each column in our sample rectangle. Hence, the multi-scale property of the wavelet transform analyzes the coefficients at different resolutions in order to extract a predominant pattern of the boundary across the different scales. If there is enough

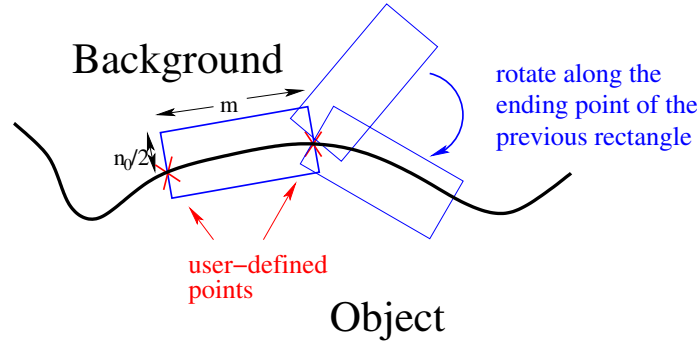


Figure 6.2: In the search for a next rectangle, a ‘candidate’ is rotated along the ending point of the previous rectangle until the *characteristic pattern* derived in step 3 is approximated best.

correspondence within neighboring columns, a *characteristic pattern* for the selected boundary is derived from the coefficients in the time–scale domain.

4. In the search for a continuation of the sample boundary, the algorithm then automatically seeks another rectangle of the same size which starts at the ending point of its predecessor and which has a similar pattern of coefficients in the time–scale domain as the characteristic pattern (see Figure 6.2). The ending point of the predecessor is used as the center of a rotation: Each angle ϕ is processed with steps 2 and 3 until a best approximation is given in the sense that the wavelet–transformed coefficients *on each scale* differ the least from the characteristic pattern of the sample rectangle.

Mathematically speaking, we want to solve the following minimization problem. Let $\text{WT}_{\text{sample}}^j(i)$ denote the wavelet transform of column j of the sample rectangle in scale i , and $\text{WT}_{\text{candidate}}^{\phi,j}(i)$ denote the wavelet transform of column j of the candidate rectangle with angle ϕ . The wavelet transform provides information about the scales $n_0/2 = 2^{n-1}, n_0/4 = 2^{n-2}, \dots, 2$. Then, the selected new rectangle with angle ϕ_0 solves

$$\sum_{j=1}^m \sum_{k=1}^{n_0-1} \frac{1}{2^k} \left(\text{WT}_{\text{sample}}^j(2^k) - \text{WT}_{\text{candidate}}^{\phi,j}(2^k) \right)^2 \longrightarrow \min,$$

where the weighting parameter $\frac{1}{2^k}$ has been introduced to equalize the weight of the different scales.

5. Store the optimal candidate rectangle with angle ϕ_0 , rename it sample rectangle, and proceed with step 2.

Observation 6.1 *It is useful to evaluate the significance of the coefficients of a specific scale for the sample rectangle. Consider e.g. a boundary blurred by high–frequency noise. On a fine scale, the coefficients in the time–scale domain would appear random (cf. the discussion of noise in Chapter 5), while the coefficients on coarser scales might not be influenced. Therefore, the user should use a sample that has a continuous structure as sample rectangle.*

Due to the multiscale analysis of our semiautomatic segmentation algorithms, it is possible to track not only ‘sharp’, but also ‘fuzzy’ borders as long as no obvious change in resolution is apparent. This property makes our algorithm especially eligible for the segmentation of blurry objects.

6.2.3 Implementation

The presented wavelet-based segmentation algorithm has been implemented in Visual C++ for a 32-bit Windows environment. The program allows to open a 24-bit `bmp` file of the local platform into a *working space*, where the user interacts with the help of the mouse [HSE00]. Two possibilities are offered to define a sample boundary:

- By pressing the *left* mouse button in the working space, the user defines a starting point on the sample rectangle, and releasing the button defines the ending point.
- If the user presses the *right* mouse button in the working space, the trace of the mouse is sampled until the button is released. A manual deletion option for a sampled boundary is also implemented.

Figure 6.3 demonstrates a result of the wavelet-based semiautomatic segmentation process for the ‘fuzzy’ boundary of the *Ostrich*’s neck. For the screenshot, our algorithm was initiated at the left lower part of the neck. The algorithm works fine for the varying background color and along the beak. It encounters problems, however, on the right lower part of the neck when the background color changes from dark to grayish, simultaneously to a change of the neck’s border hue from bright into dark. A stable tracking at this position would require the further intelligence mentioned in Section 6.2.1.



Figure 6.3: Example for semiautomatic segmentation.

6.2.4 Experimental Results

We have evaluated our algorithm against two other semiautomatic segmentation methods (see Section 6.2.1). The results have been presented in [HSE00].

- *Edge-guided line trace*: The user defines a closed polygon that fully encloses the object. For each composing line of the polygon, the algorithm searches the strongest edge in a pre-defined neighborhood of the line. This leads to a contraction of the object-defining polygon.
- *Region-growing*: The user defines a starting point within the object. Based on the histogram of the image, the algorithm fills in the area around the starting point where the value of the pixel difference is below a certain threshold. The filled-in area is considered as a part of the object. Usually, the user has to define several object points until the result is satisfactory.

All three segmentation methods were applied to the images *Sea*, *Africa*, *Fashion*, and *Noise* (see Figure 6.4). The results of the subjective evaluation are presented in Table 6.1 [HSE00]. For the evaluation of segmentation algorithms, Mortensen [MB95] has measured the average time needed to complete the segmentation. Our experience is that the time needed depends strongly on the user's familiarity with the topic as well as with the specific tool. Therefore we decided to base our evaluation on the number of interactions required. The subjective *quality* of the segmentation result has then been evaluated on a scale from 1 (very poor) to 10 (excellent).

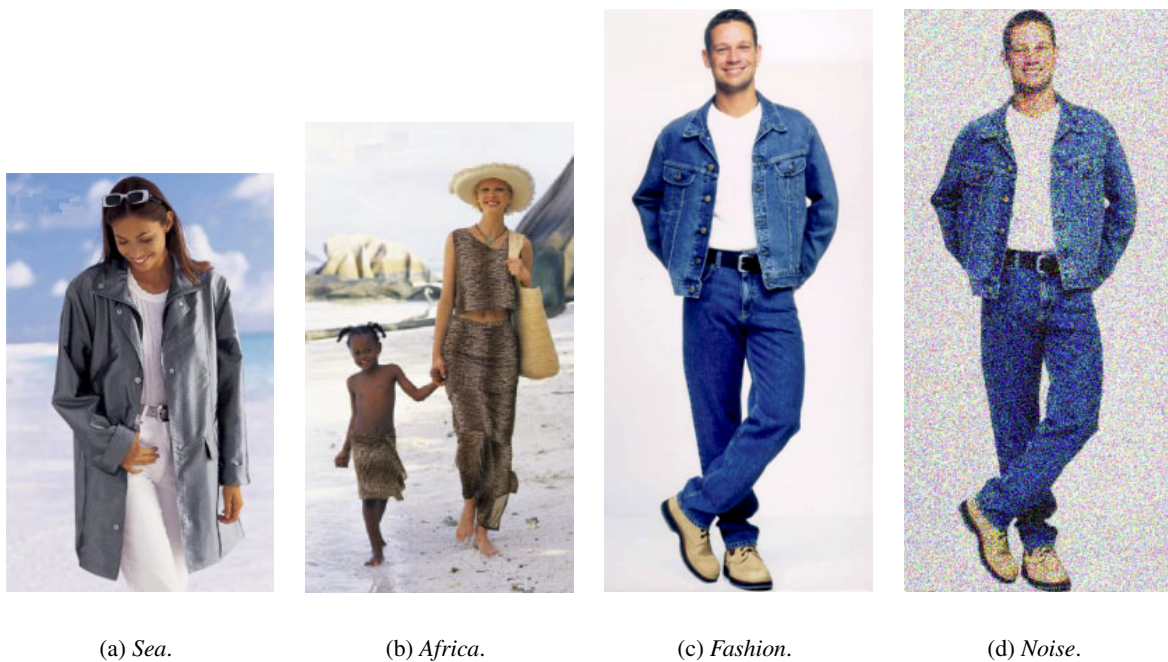


Figure 6.4: Test images for the empirical evaluation of the different segmentation algorithms.

During this test, the segmentation process was terminated when the improvement of the image's quality was so marginal that it would not justify further user interaction. From Table 6.1, it can be seen that the best results were achieved with either the edge-guided or our semiautomatic wavelet-based segmentation. In a next step, we included the number of interactions into our quality considerations. Table 6.2 shows an evaluation of the different segmentation algorithms when the effort to achieve a given segmentation quality was taken into consideration. The column 'average quality' of Table 6.2 shows the sum of the quality results on the four images for each segmentation method. Similarly,

the column ‘# user interactions’ gives the sum of the interactions per method over the four images. The ‘overall quality’ is the relation of ‘quality per effort’. Measured on this scale, the presented segmentation algorithm clearly outperformed its competitors in the overall evaluation.

Method	# Interact.	Test Person							Average quality
		P1	P2	P3	P4	P5	P6	P7	
Sea									
Edge-guided	28	4	6	3	4	6	5	6	4.85
Region-growing	40	2	3	2	3	3	1	5	2.71
Wavelet-based	21	8	8	7	10	9	8	9	8.43
Africa									
Edge-guided	102	8	5	6	7	7	5	8	6.57
Region-growing	57	2	3	2	3	2	1	4	2.43
Wavelet-based	59	3	4	5	5	3	5	6	4.43
Fashion									
Edge-guided	55	4	5	5	8	5	6	8	5.86
Region-growing	2	6	7	7	8	8	8	8	7.43
Wavelet-based	2	9	9	9	9	10	10	9	9.29
Noise									
Edge-guided	60	6	6	5	8	5	7	8	6.43
Region-growing	4	1	2	2	3	1	1	3	1.86
Wavelet-based	2	10	8	8	9	9	10	10	9.14

Table 6.1: Experimental results for three different segmentation algorithms. Here, the semiautomatic wavelet-based segmentation was implemented with the Haar filter bank. The subjective quality was rated by seven test persons on a scale from 1 (very poor) to 10 (excellent).

Although the empirical evaluation was not carried out on a sample large enough to allow a general statement, we have demonstrated the feasibility of the idea to apply the multiscale property of the wavelet transform to image segmentation, and we have demonstrated the power of this new approach.

Method	Average quality	# User interactions	Overall quality
Edge-guided	23.71	245	0.0968
Region-growing	14.43	103	0.1401
Wavelet-based	31.29	84	0.3715

Table 6.2: Experimental results: summary of the four test images for three different segmentation algorithms. The overall quality is the relation of perceived subjective quality to the number of interactions.

6.3 Empirical Parameter Evaluation for Image Coding

In Section 3.3, we have discussed the implementation problems of a wavelet transform that occur on the signal boundary. We have presented the two main types of boundary treatment: circular convolution and padding (see Sections 3.3.1 and 3.3.2), and we have deduced that the choice of boundary treatment has an important impact on the iteration behavior of the wavelet transform (see Section 3.3.3).

An important aspect of all wavelet-based applications is the answer to the question, *Which wavelet filter bank shall be used for the specific problem?*

The short-time Fourier transform occupies an easy position in this debate: Its basis function in the transformed space is the exponential function that decomposes into \cos for the real part and \sin for the imaginary part. The discrete cosine transform (see Sections 9.3 and 9.4), which is the transform underlying the coding standard JPEG [PM93], leaves no choice either: Its basis function in the transformed space is restricted to \cos , which is real-valued and thus very convenient in signal processing applications. The wavelet transform, however, has the great inconvenience for implementors that a wavelet is not a specific function, but a whole class of functions.

Research has been carried out on this question since the early days of wavelets, and Daubechies says that there is no best answer in general. It all depends on the specific problem as well as on the specific purpose.

Some research groups have carried out wavelet filter evaluations: Villasenor's group researches wavelet filters for image compression. In [VBL95], the focus is on biorthogonal filters, and the evaluation is based on the information preserved in the reference signal, while [GFBV97] focuses on a mathematically optimal quantizer step size. In [AK99], the evaluation is based on lossless as well as on subjective lossy compression performance, complexity, and memory usage.

We have taken a different direction in the empirical parameter evaluation that is presented in this section. In a first evaluation (see Section 6.3.2), we were interested in the performance of different boundary policies for the wavelet transform. A second evaluation (see Section 6.3.3) takes a closer look at the performance of different orthogonal Daubechies wavelet filter banks and answers the question of which filter to use for image coding. A third evaluation (see Section 6.3.4) discusses the impact of the selected decomposition strategy for best parameter settings. These results have been presented at [Sch01b], [Sch01c], and [Sch02].

6.3.1 General Setup

The goal of our empirical evaluation was to find the best parameter settings for wavelet transforms of still images: The image *boundary policy*, the *choice of the wavelet filter bank*, and the *decomposition strategy* of the separable two-dimensional wavelet transform. The overall performance was evaluated according to the three criteria:

1. visual quality,
2. compression rate, and

3. complexity of implementation.

The rating of the visual quality of the decoded images was based on the *peak signal-to-noise ratio* (PSNR)¹. The compression rate was simulated by a simple quantization threshold: The higher the threshold, the more coefficients in the time-scale domain are discarded, and the higher is the compression rate. Four quantization thresholds (i.e., $\lambda = 10, 20, 45, 85$) were selected which present heuristics for excellent, good, medium, and poor coding quality.

Our evaluation was set up on the six grayscale images of size 256×256 pixels shown in Figure 6.7 in Section 6.3.6. These test images were selected since they constitute a good data base and comply with the following features:

- contain many small details: *Mandrill, Goldhill,*
- contain large uniform areas: *Brain, Lena, Camera, House,*
- are relatively symmetric at the left-right and top-bottom boundaries: *Mandrill, Brain,*
- are very asymmetric with regard to these boundaries: *Lena, Goldhill, House,*
- have sharp transitions between regions: *Brain, Lena, Camera, House,*
- contain large areas of texture: *Mandrill, Lena, Goldhill, House.*

The orthogonal and separable wavelet filters that Daubechies [Dau92] has developed compose the group of wavelets used most often in image coding applications (see Section 1.3.2.4); we have concentrated on this wavelet class. The number n_0 of vanishing moments of a Daubechies wavelet specifies the approximation order of the wavelet transform. A fast approximation is mathematically desirable. However, the filter length has an impact on the cost of calculation as well as on image quality.

In the following, we present the three empirical evaluations of the performance of different boundary policies, of the performance of different orthogonal Daubechies wavelet filter banks, and of the impact of the selected decomposition strategy.

6.3.2 Boundary Policies

In this section, we investigate different wavelet filter banks in combination with different boundary policies [Sch02]. When *circular convolution* is chosen as the boundary treatment, the level of iteration depends on the length of the selected filter bank (see Section 3.3.3). While the level of iterations of the transform thus decreases with increasing filter length for circular convolution, the test images of size 256×256 pixels have been decomposed into 8 levels with padding policies.

¹When $\text{org}(x, y)$ depicts the pixel value of the original image at position (x, y) , and $\text{dec}(x, y)$ denotes the pixel value of the decoded image at position (x, y) , then $\text{PSNR [dB]} = 10 \cdot \log_{10} \left(\frac{\sum_{x,y} 255^2}{\sum_{x,y} (\text{org}(x, y) - \text{dec}(x, y))^2} \right)$, where the value 255 enters the formula as the maximum possible difference in pixel value (thus, *peak*).

In signal analysis it is extremely difficult to empirically derive general statements as results usually depend on the signal under consideration. Therefore, we present both an image-dependent analysis as well as an image-independent analysis, based on the obtained mean values.

We focus on the question of whether the circular convolution — in contrast to padding — with its ease of implementation and its drawback on the number of iteration levels, and its thus restricted potentiality to concentrate the signal's energy within a few large coefficients, provokes any deteriorated quality in the decoded image.

6.3.2.1 Image-dependent Analysis

The detailed image-dependent evaluation results for the six test images in Figure 6.7 are presented in Tables 6.3 and 6.4 in Section 6.3.6. Table 6.3 lists the image quality, measured in decibels (dB) for each of the six test images when the three parameters

- padding policy: zero padding, mirror padding, circular convolution,
- wavelet filter bank: Daub-2, Daub-3, Daub-4, Daub-5, Daub-10, Daub-15, Daub-20, and
- quantization threshold: $\lambda = 10, 20, 45, 85$,

were varied. Table 6.4 varied the same parameters as well, but rather than measuring the image quality, it presents the compression rate at a given parameter setting. This was obtained by measuring the amount of discarded coefficients in the time-scale domain: the higher the percentage of discarded information, the higher the compression ratio. Some interesting observations are:

- For a given image and a given quantization threshold, the PSNR remains astonishingly constant for different filter banks and different boundary policies.
- At high thresholds, *Mandrill* and *Goldhill* yield the worst quality. This is due to the large amount of details in both images.
- *House* delivers the overall best quality at a given threshold. This is due to its large uniform areas.
- Due to their symmetry, *Mandrill* and *Brain* show good-quality results for padding policies.
- The percentage of discarded information at a given threshold is far higher for *Brain* than for *Mandrill*. This is due to the uniform black background of *Brain*, which produces small coefficients in the time-scale domain, compared to the many small details in *Mandrill*, which produce large coefficients and thus do not fall below the threshold.
- With regard to the compression rate, and for a given image and filter bank, Table 6.4 reveals that
 - the compression ratio for zero padding *increases* with increasing filter length,
 - the compression ratio for mirror padding *decreases* with increasing filter length, and

- the compression ratio for circular convolution varies, but most often remains *almost constant*.

The explanation for the latter observation is as follows. Padding an image with zeros, i.e., black pixel values, most often creates a sharp contrast to the original image; thus the sharp transition between the signal and the padding coefficients results in large coefficients in the fine scales, while the coarse scales remain unaffected. This observation, however, is put into a different perspective for longer filters: With longer filters, the constant run of zeros at the boundary forces the detail coefficients in the time–scale domain to remain small. Hence, a given threshold cuts off fewer coefficients when the filter is longer.

With mirror padding, the padded coefficients for shorter filters represent a good heuristic for the signal near the boundary. Increasing filter length and accordingly longer padded areas, however, introduce too much ‘false’ detail information into the signal, resulting in many large detail coefficients that ‘survive’ the threshold.

The following sections discuss so many parameter combinations that it is impossible to present a visual example (i.e., a screenshot) for every parameter setting. Figures 6.8 to 6.11, however, elucidate the impact of the threshold ($\lambda = 10, 20, 45, 85$) on the test images with a Daub–5 filter bank, circular convolution as the boundary policy, and standard decomposition.

6.3.2.2 Image-independent Analysis

Our further reflections are made on the *average* image quality and the *average* amount of discarded information as presented in Tables 6.5 and 6.6 and the corresponding Figures 6.12 and 6.13.

Figure 6.12 visualizes the coding quality of the images, averaged over the six test images. The four plots represent the quantization thresholds $\lambda = 10, 20, 45, 85$. In each graphic, the visual quality (quantified via PSNR) is plotted against the filter length of the Daubechies wavelet filters. The three boundary policies *zero padding*, *mirror padding*, and *circular convolution* are regarded separately. The plots obviously reveal that the quality decreases with an increasing threshold. More important are the following statements:

- Within a given threshold, and for a given boundary policy, the PSNR remains almost constant. This means that the quality of the coding process either does not or hardly depends on the selected wavelet filter bank.
- Within a given threshold, mirror padding produces the best results, followed by those for circular convolution. Zero padding performs worst.
- The gap between the performance of the boundary policies increases with an increasing threshold.

Nevertheless, the differences observed above with 0.28 dB maximum gap (at $\lambda = 85$ and filter length = 40) are so marginal that they do not actually influence the visual perception.

As the visual perception is neither influenced much by the choice of filter nor by the boundary policy, the compression rate has been studied as a second benchmark (see Section 6.3.1). The following observations are made from Figure 6.13. With a short filter length (4 to 10 taps), the compression ratio is almost identical for the different boundary policies. This is not surprising, as short filters involve only little boundary treatment, and the relative importance of the boundary coefficients with regard to the signal coefficients is negligible. More important for our investigation are:

- The compression rate for each of the three boundary policies is inversely proportional to their quality performance. In other words, mirror padding discards the least number of coefficients at a given quantization threshold, while zero padding discards the most.
- The compression ratio for mirror padding worsens with an increasing filter length and thus with an increasing number of padded coefficients. However, it remains almost constant for circular convolution, and slightly improves for zero padding.
- With an increasing threshold, the gaps between the compression ratios of the three policies narrow.

In the overall evaluation, we have shown that mirror padding performs best with regard to quality, while it performs worst with regard to compression. Inversely, zero padding performs best with regard to compression and worst with regard to quality. Circular convolution holds the midway in both aspects. On the other hand, the difference in compression rates is by far superior to the difference in quality. If we now call to mind the coding complexity of the padding approaches, compared to the ease of implementation of circular convolution (see Section 3.3), we strongly recommend to implement circular convolution as the boundary policy in image coding.

6.3.3 Choice of Orthogonal Daubechies Wavelet Filter Bank

Tables 6.3 and 6.4 not only reveal a best practice for boundary treatment, but also contain information about a best choice of wavelet filter bank. In Table 6.3, for each selection of image, threshold, and boundary policy, the filter bank with the best visual perception is marked in bold face.

The evaluation shows that the exact definition of a best-suited wavelet filter bank depends on the selection of the other relevant parameters. However, most often, a medium-length filter bank wins the race [SKE01b].

This observation finds its theoretical explanation in the fact that the short filter banks are still too irregular and thus their artifacts at poor quality very much disturb visual perception. The longer filters, however, require a greater number of boundary coefficients on the one hand, while on the other their impact of one coefficient in the time-scale domain involves many more coefficients of the original image. This leads to a very ‘flickery’ image at poor quality which usually also disturbs visual perception. Figure 6.5 shows the impact of different wavelet filter banks on the visual perception of a strongly compressed image when all other parameters have been set identically. Figure 6.5 (a) has a PSNR of 12.213 and the quality subjectively appears much superior to that of Figure 6.5 (b) which has a PSNR of 11.798. Note, this result is independent of the size of the image.



(a) Daub-2 filter bank: PSNR=12.213.



(b) Daub-20 filter bank: PSNR=11.798.

Figure 6.5: Impact of different wavelet filter banks on visual perception. Both images were coded with zero padding as the boundary policy, nonstandard decomposition, and a threshold of $\lambda = 45$.

Concerning the choice of wavelet filter bank, we thus recommend filters of medium length (Daub-5 with 10 taps to Daub-10 with 20 taps), as their overall coding quality is superior to both shorter and longer filter banks.

6.3.4 Decomposition Strategies

In Section 3.2, we have stated that the separable approach of the wavelet transform on still images allows two kinds of decomposition, standard and nonstandard. Interestingly enough, the overwhelming majority of current research concentrates on the nonstandard decomposition. This is also true for JPEG2000.

As the standard decomposition allows a more finely grained subdivision into approximations and details (see Section 3.2), we were interested whether this could be successfully exploited for low-bit rate coding [Sch01b]. Analogous to the setup in the previous sections, the empirical evaluation of the performance of the two decomposition strategies was rated based on the perceived visual quality, measured with the PSNR. Since the evaluation in Section 6.3.2 suggests to implement circular convolution, we have concentrated on this boundary policy. Hence, the iteration depth again depends on the length of the filter bank.

Figure 6.6 gives an impression of the visual difference resulting from a variation in the decomposition policy. The parameters were set to the Daub-20 wavelet filter bank, circular convolution, and a threshold of $\lambda = 85$. Figure 6.6 (a) was coded with the standard decomposition and Figure 6.6 (b) with the nonstandard decomposition. The visual perception of both images is very close; as is the PSNR: 11.119 in (a) and 11.090 in (b). Apart from this example, the results of the empirical evaluation are given in Table 6.7. The values of the nonstandard decomposition in Table 6.7 correspond to

the columns ‘circular convolution’ in Table 6.3. We have included them again in order to allow a direct comparison of the visual quality of both decomposition strategies. However, the values for the standard decomposition are new.



(a) standard decomposition: PSNR = 11.119.



(b) nonstandard decomposition: PSNR = 11.090.

Figure 6.6: Impact of different decomposition strategies on visual perception. Both images were coded with the Daub-20 wavelet filter bank, circular convolution, and a threshold of $\lambda = 85$.

Table 6.8 shows the average values over the six test images. For better visualization, we have included the filter length (i.e., the number of taps) and the iteration depth with the respective filter bank in this table. Figure 6.14 is the corresponding plot. This evaluation shows again that the quality of both decomposition strategies is astonishingly similar: The maximum difference is 0.188 dB (with $\lambda = 10$ and the Daub-5 filter bank), and the average difference is 0.064 dB.

However, we state a superiority of the nonstandard decomposition at good coding quality, while the performance by standard decomposition is superior at poor quality (i.e., at low bit rates). This is due to the rigorous separation of details and approximations of the standard decomposition in the mixed terms (see Equations (3.3) and (3.4)).

6.3.5 Conclusion

We have discussed and evaluated the strengths and weaknesses of different parameter settings in a separable two-dimensional wavelet transform with regard to the boundary policy, the choice of the Daubechies filter bank, and the decomposition strategy.

We have revealed that within a given quality threshold and for a given image the visual perception of most parameter settings is astonishingly similar. Big differences, however, can be stated for multiple images. The analysis of the mean values over our six test images nevertheless allows the conclusion that, in general, the following three statements hold:

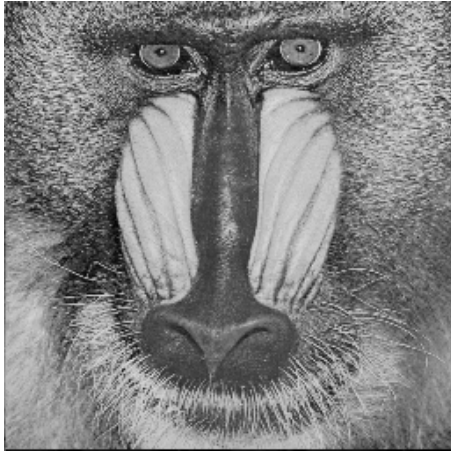
1. An orthogonal wavelet filter bank of medium length is the best trade-off between the regularity

of the transformation and the expansion of disturbing artifacts.

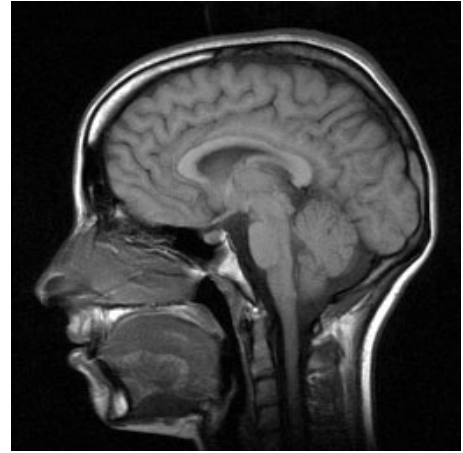
2. The coding quality depends on the boundary policy selected, and *mirror padding* generally produces the best results. Nevertheless, the difference is not significant (see Section 6.3.2.2). The average bit rate of the three policies reveals that all three perform comparably for shorter wavelet filters, while *zero padding* thresholding affects a larger share of the coefficients when the filter length increases. Since medium length wavelet filters produce better visual quality (see argument 1.), this difference becomes less important, and it is the coding complexity that ultimately decides the competition. Here, *circular convolution* is superior, thus it represents the best trade-off between coding quality, compression rate, and coding complexity.
3. In low-bit rate coding, the standard decomposition qualitatively outperforms the nonstandard decomposition suggested for JPEG2000.

6.3.6 Figures and Tables of Reference

The following pages show the test images, the tables, and the plots discussed in the above sections.



(a) *Mandrill.*



(b) *Brain.*



(c) *Lena.*



(d) *Camera.*



(e) *Goldhill.*



(f) *House.*

Figure 6.7: Test images for the empirical parameter evaluation: grayscale, 256×256 pixels.

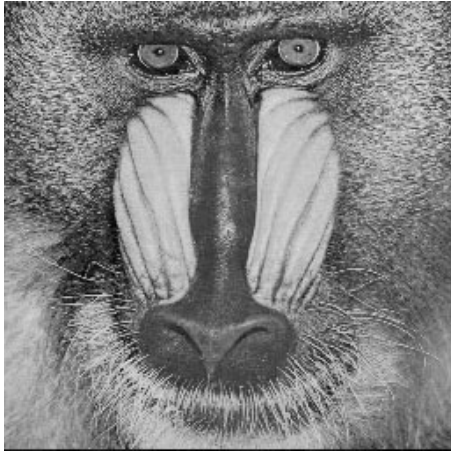
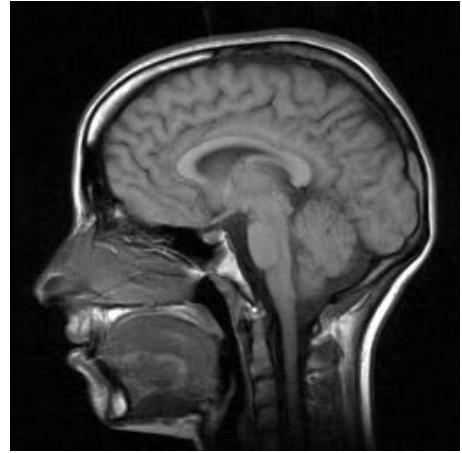
(a) *Mandrill.*(b) *Brain.*(c) *Lena.*(d) *Camera.*(e) *Goldhill.*(f) *House.*

Figure 6.8: Test images with threshold $\lambda = 10$ in the time-scale domain with Daub-5 filter bank, circular convolution, and standard decomposition.

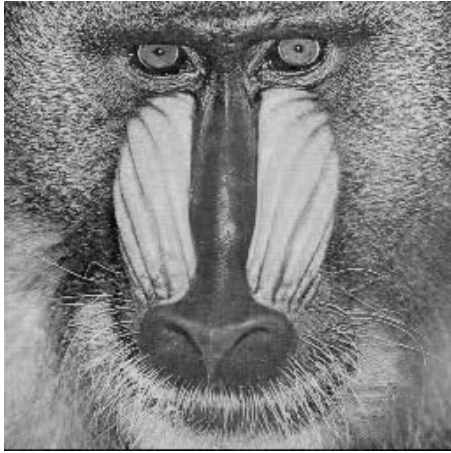
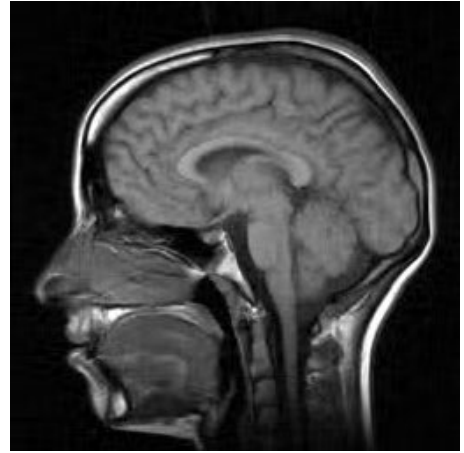
(a) *Mandrill.*(b) *Brain.*(c) *Lena.*(d) *Camera.*(e) *Goldhill.*(f) *House.*

Figure 6.9: Test images with threshold $\lambda = 20$ in the time-scale domain with Daub-5 filter bank, circular convolution, and standard decomposition.

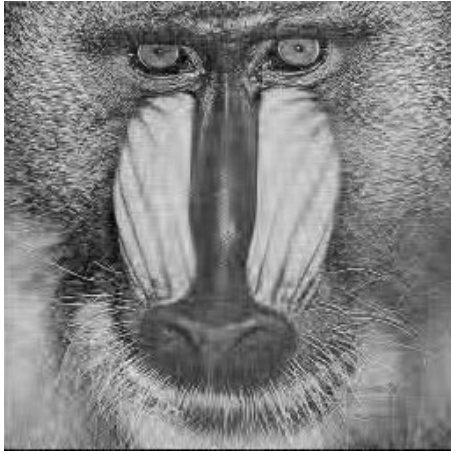
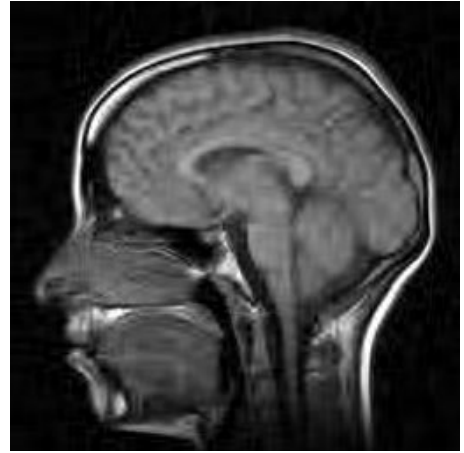
(a) *Mandrill.*(b) *Brain.*(c) *Lena.*(d) *Camera.*(e) *Goldhill.*(f) *House.*

Figure 6.10: Test images with threshold $\lambda = 45$ in the time-scale domain with Daub-5 filter bank, circular convolution, and standard decomposition.

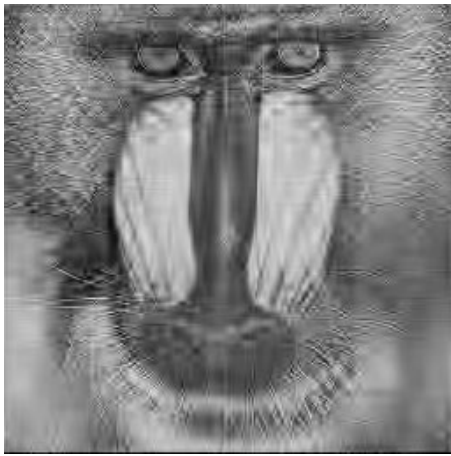
(a) *Mandrill.*(b) *Brain.*(c) *Lena.*(d) *Camera.*(e) *Goldhill.*(f) *House.*

Figure 6.11: Test images with threshold $\lambda = 85$ in the time-scale domain with Daub-5 filter bank, circular convolution, and standard decomposition.

Quality of visual perception — PSNR [dB]									
Wavelet	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.
	Mandrill			Brain			Lena		
	Threshold $\lambda = 10$ — Excellent overall quality								
Daub-2	18.012	17.996	18.238	18.141	18.151	18.197	16.392	16.288	16.380
Daub-3	18.157	18.187	18.221	18.429	18.434	18.433	16.391	16.402	16.350
Daub-4	18.169	18.208	17.963	18.353	18.340	18.248	16.294	16.355	16.260
Daub-5	18.173	18.167	18.186	18.279	18.280	18.259	16.543	16.561	16.527
Daub-10	17.977	17.959	18.009	18.291	18.300	18.479	16.249	16.278	16.214
Daub-15	17.938	17.934	18.022	18.553	18.543	18.523	16.267	16.304	16.288
Daub-20	17.721	17.831	18.026	18.375	18.357	18.466	16.252	16.470	16.238
	Threshold $\lambda = 20$ — Good overall quality								
Daub-2	14.298	14.350	14.403	16.610	16.611	16.577	14.775	14.765	14.730
Daub-3	14.414	14.469	14.424	16.743	16.755	16.721	14.758	14.817	14.687
Daub-4	14.231	14.239	14.276	16.637	16.628	16.734	14.862	14.918	14.735
Daub-5	14.257	14.216	14.269	16.747	16.751	16.854	14.739	14.946	14.815
Daub-10	14.268	14.274	14.360	16.801	16.803	16.878	14.624	14.840	14.699
Daub-15	14.246	14.258	14.300	16.822	16.810	16.852	14.395	14.631	14.477
Daub-20	14.046	14.065	14.227	16.953	16.980	16.769	14.252	14.597	14.353
	Threshold $\lambda = 45$ — Medium overall quality								
Daub-2	10.905	10.885	10.910	14.815	14.816	14.747	13.010	13.052	12.832
Daub-3	10.988	10.970	10.948	15.187	15.150	15.052	12.766	13.138	12.903
Daub-4	10.845	10.839	10.885	15.014	15.029	15.056	12.820	13.132	12.818
Daub-5	10.918	10.969	10.949	15.036	15.031	14.999	12.913	13.301	12.983
Daub-10	10.907	10.929	10.913	14.989	15.013	15.212	12.447	13.066	12.795
Daub-15	10.845	10.819	10.815	15.093	15.133	15.064	12.577	12.954	12.686
Daub-20	10.784	10.872	10.843	14.975	14.934	14.882	12.299	12.877	12.640
	Threshold $\lambda = 85$ — Poor overall quality								
Daub-2	9.095	9.121	9.135	13.615	13.621	13.783	11.587	11.902	11.577
Daub-3	9.206	9.184	9.124	13.787	13.784	13.759	11.437	11.793	11.516
Daub-4	9.160	9.152	9.168	13.792	13.815	13.808	11.539	11.806	11.636
Daub-5	9.171	9.208	9.203	13.837	13.850	13.705	11.692	11.790	11.872
Daub-10	9.207	9.193	9.206	13.870	13.922	14.042	11.128	11.430	11.555
Daub-15	9.083	9.161	9.126	13.731	13.795	13.917	11.128	11.610	11.475
Daub-20	9.071	9.142	9.204	13.852	13.800	13.974	11.142	11.694	11.597
	Camera			Goldhill			House		
	Threshold $\lambda = 10$ — Excellent overall quality								
Daub-2	17.334	17.346	17.371	16.324	16.266	16.412	19.575	19.563	19.608
Daub-3	17.532	17.560	17.625	16.322	16.296	16.358	19.640	19.630	19.621
Daub-4	17.529	17.591	17.577	16.241	16.212	16.342	19.560	19.558	19.584
Daub-5	17.489	17.448	17.389	16.214	16.193	16.154	19.613	19.555	19.566
Daub-10	17.539	17.541	17.383	16.307	16.223	16.317	19.482	19.388	19.732
Daub-15	17.747	17.530	17.523	16.012	16.067	16.033	19.653	19.671	19.726
Daub-20	17.474	17.527	17.484	16.322	16.245	16.319	19.550	19.495	19.524
	Threshold $\lambda = 20$ — Good overall quality								
Daub-2	14.387	14.365	14.396	13.937	13.940	13.898	17.446	17.480	17.471
Daub-3	14.473	14.452	14.426	13.872	13.892	13.858	17.525	17.594	17.612
Daub-4	14.438	14.438	14.430	13.828	13.836	13.753	17.468	17.647	17.351
Daub-5	14.460	14.505	14.427	13.743	13.743	13.711	17.454	17.458	17.465
Daub-10	14.468	14.400	14.409	13.762	13.785	13.798	17.592	17.635	17.689
Daub-15	14.408	14.406	14.414	13.687	13.730	13.697	17.260	17.276	17.266
Daub-20	14.384	14.370	14.362	13.700	13.782	13.731	17.476	17.449	17.240
	Threshold $\lambda = 45$ — Medium overall quality								
Daub-2	12.213	12.242	12.131	12.033	12.034	11.876	15.365	15.437	15.155
Daub-3	12.032	12.122	12.188	11.961	12.006	11.889	14.957	15.476	15.118
Daub-4	12.150	12.178	12.145	11.855	11.891	11.925	14.906	15.080	15.180
Daub-5	12.077	12.133	12.120	11.848	11.844	11.801	15.159	15.382	15.244
Daub-10	12.061	12.197	12.093	11.760	11.917	11.726	14.776	15.246	14.872
Daub-15	12.074	12.059	12.176	11.725	11.855	11.753	14.810	15.090	14.969
Daub-20	11.798	11.975	12.048	11.763	11.803	11.703	14.420	15.033	14.609
	Threshold $\lambda = 85$ — Poor overall quality								
Daub-2	11.035	11.161	11.041	10.791	10.805	10.844	13.530	13.804	13.703
Daub-3	11.092	11.176	11.080	10.943	10.916	10.754	13.488	13.726	13.627
Daub-4	10.943	11.152	11.046	10.861	10.904	10.740	13.524	13.613	13.510
Daub-5	11.018	11.148	11.129	10.826	10.935	10.738	13.114	13.903	13.111
Daub-10	10.815	11.064	10.987	10.824	10.972	10.771	13.158	13.695	13.434
Daub-15	10.779	11.005	10.982	10.737	10.838	10.607	13.073	13.357	13.123
Daub-20	10.688	11.031	11.090	10.709	10.819	10.766	13.173	13.257	13.678

Table 6.3: Detailed results of the quality evaluation for the six test images. The mean values over the images for a fixed wavelet filter bank and a fixed boundary policy are given in Table 6.5.

Discarded information in the time-scale domain due to the threshold — Percentage [%]									
Wavelet	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.
	Mandrill			Brain			Lena		
	Threshold $\lambda = 10$ — Excellent overall quality								
Daub-2	42	41	41	83	83	83	78	79	79
Daub-3	43	42	42	84	84	84	78	80	80
Daub-4	44	42	41	85	84	84	78	79	79
Daub-5	45	41	41	85	84	84	79	79	80
Daub-10	53	38	41	87	82	84	79	74	78
Daub-15	59	35	40	88	78	82	82	69	77
Daub-20	65	32	40	89	74	83	83	64	77
	Threshold $\lambda = 20$ — Good overall quality								
Daub-2	63	63	63	91	91	91	87	89	88
Daub-3	64	63	64	92	91	91	87	89	89
Daub-4	65	63	63	92	91	91	87	88	89
Daub-5	66	62	63	92	91	91	87	90	89
Daub-10	70	58	63	93	89	91	88	83	88
Daub-15	74	56	62	93	86	91	89	79	88
Daub-20	78	51	63	94	82	91	90	74	88
	Threshold $\lambda = 45$ — Medium overall quality								
Daub-2	86	86	87	96	96	96	94	95	95
Daub-3	86	86	87	96	96	96	94	95	95
Daub-4	87	86	87	96	96	96	94	95	96
Daub-5	87	85	87	96	96	96	95	94	96
Daub-10	88	82	87	97	94	96	94	91	96
Daub-15	90	79	87	97	91	96	95	88	96
Daub-20	92	74	87	97	89	96	96	83	96
	Threshold $\lambda = 85$ — Poor overall quality								
Daub-2	96	96	97	98	98	98	97	98	98
Daub-3	96	96	97	98	98	98	97	98	98
Daub-4	96	96	97	98	98	98	97	97	98
Daub-5	96	95	97	98	98	98	98	97	98
Daub-10	97	93	97	98	97	98	97	94	98
Daub-15	97	91	97	98	95	98	98	92	98
Daub-20	97	86	98	98	93	99	98	88	99
	Camera			Goldhill			House		
	Threshold $\lambda = 10$ — Excellent overall quality								
Daub-2	78	80	79	70	71	70	79	80	80
Daub-3	77	79	78	70	71	71	79	80	80
Daub-4	77	79	78	71	71	70	79	80	79
Daub-5	77	78	78	71	71	70	79	79	79
Daub-10	77	74	76	73	67	69	80	72	78
Daub-15	80	71	75	77	63	68	82	66	77
Daub-20	81	66	74	79	58	68	83	59	76
	Threshold $\lambda = 20$ — Good overall quality								
Daub-2	86	88	88	85	87	86	87	88	88
Daub-3	86	88	88	85	87	86	87	88	88
Daub-4	86	88	88	86	86	86	87	88	87
Daub-5	86	87	88	86	86	86	87	87	88
Daub-10	86	85	87	86	83	86	87	81	87
Daub-15	88	82	86	89	79	86	89	75	87
Daub-20	88	78	86	89	73	86	89	69	87
	Threshold $\lambda = 45$ — Medium overall quality								
Daub-2	93	95	95	94	96	95	93	95	94
Daub-3	93	95	95	95	96	95	94	95	95
Daub-4	94	95	95	95	95	95	94	94	95
Daub-5	94	94	95	95	95	96	94	94	95
Daub-10	93	93	95	95	92	96	94	89	95
Daub-15	94	91	95	95	89	96	95	84	94
Daub-20	95	88	95	96	85	96	95	78	95
	Threshold $\lambda = 85$ — Poor overall quality								
Daub-2	97	98	98	97	98	98	97	98	98
Daub-3	97	98	98	98	98	98	97	97	97
Daub-4	97	98	98	98	98	98	97	97	98
Daub-5	97	97	98	98	98	99	97	97	98
Daub-10	97	96	98	98	96	99	97	93	98
Daub-15	97	95	98	98	93	99	97	89	98
Daub-20	98	93	98	98	90	99	98	84	99

Table 6.4: Heuristic for the compression rate of the coding parameters of Table 6.3. The mean values over the images for a fixed wavelet filter bank and a fixed boundary policy are given in Table 6.6.

Average image quality — PSNR [dB]						
Wavelet	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.
Threshold $\lambda = 10$						
Daub-2	17.630	17.602	17.701	15.242	15.252	15.246
Daub-3	17.745	17.752	17.768	15.298	15.330	15.288
Daub-4	17.691	17.711	17.662	15.244	15.284	15.213
Daub-5	17.719	17.701	17.680	15.233	15.270	15.257
Daub-10	17.641	17.615	17.689	15.253	15.290	15.306
Daub-15	17.695	17.675	17.686	15.136	15.185	15.168
Daub-20	17.616	17.654	17.676	15.135	15.207	15.114
Threshold $\lambda = 45$						
Daub-2	13.057	13.078	12.942	11.609	11.736	11.681
Daub-3	12.982	13.144	13.016	11.659	11.763	11.643
Daub-4	12.932	13.025	13.002	11.637	11.740	11.651
Daub-5	12.992	13.110	13.016	11.610	11.806	11.626
Daub-10	12.823	13.061	12.935	11.500	11.713	11.666
Daub-15	12.854	12.985	12.911	11.422	11.628	11.538
Daub-20	12.673	12.916	12.788	11.439	11.624	11.718
Threshold $\lambda = 85$						
Daub-2	13.057	13.078	12.942	11.609	11.736	11.681
Daub-3	12.982	13.144	13.016	11.659	11.763	11.643
Daub-4	12.932	13.025	13.002	11.637	11.740	11.651
Daub-5	12.992	13.110	13.016	11.610	11.806	11.626
Daub-10	12.823	13.061	12.935	11.500	11.713	11.666
Daub-15	12.854	12.985	12.911	11.422	11.628	11.538
Daub-20	12.673	12.916	12.788	11.439	11.624	11.718

Table 6.5: Average quality of the six test images. Figure 6.12 gives a more ‘readable’ plot of these numbers.

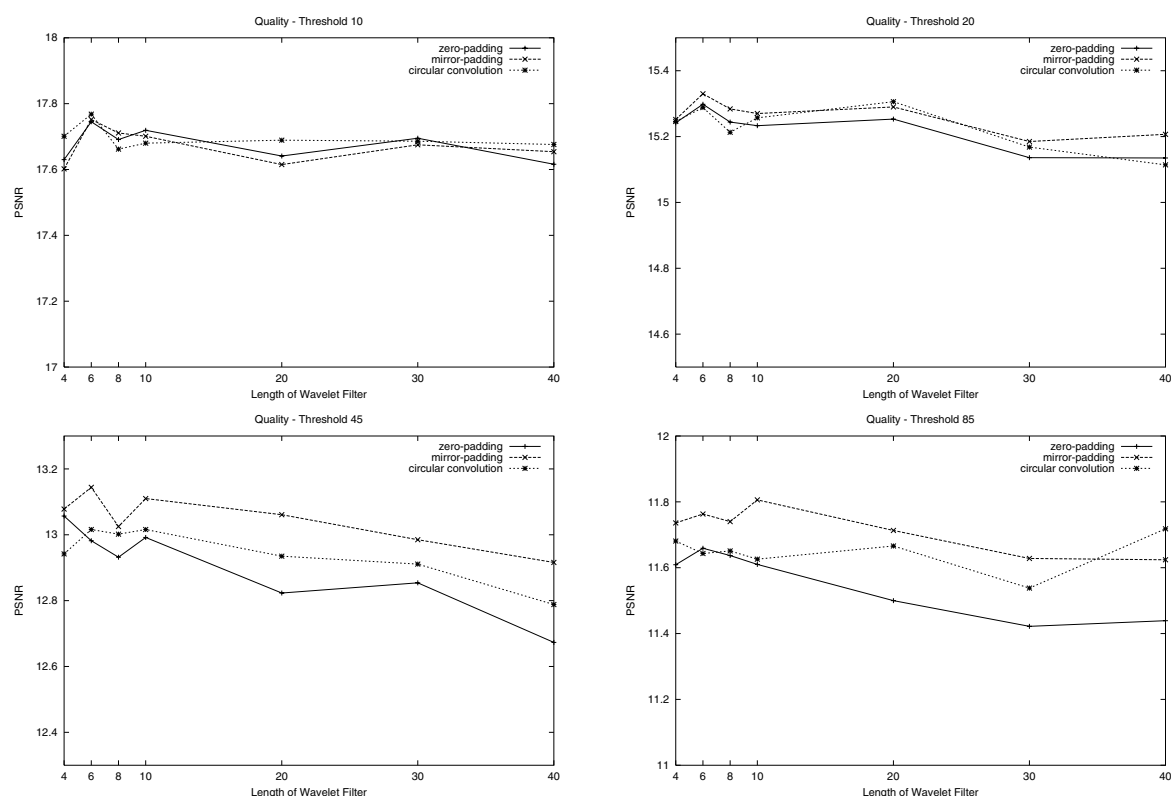


Figure 6.12: Visual quality of the test images at the quantization thresholds $\lambda = 10, 20, 45, 85$. The values are averaged over the six test images and correspond to those in Table 6.5. Each plot covers a range of one dB for the PSNR. Note that the perceived quality decreases with a decreasing PSNR.

Average discarded information — Percentage [%]						
Wavelet	zero padding	mirror padding	circular convol.	zero padding	mirror padding	circular convol.
Threshold $\lambda = 10$						
Daub-2	72.0	72.3	72.0	83.2	84.3	84.0
Daub-3	71.8	72.7	72.5	83.5	84.3	84.3
Daub-4	72.3	72.5	71.8	83.8	84.0	84.0
Daub-5	72.7	72.0	72.0	84.0	83.8	84.2
Daub-10	74.8	67.8	71.0	85.0	79.8	83.7
Daub-15	78.0	63.7	69.8	87.0	76.2	83.3
Daub-20	80.0	58.8	69.7	88.0	71.2	83.5
Threshold $\lambda = 45$						
Daub-2	92.7	93.8	93.7	97.0	97.7	97.8
Daub-3	93.0	93.8	93.8	97.2	97.5	97.7
Daub-4	93.3	93.5	94.0	97.2	97.3	97.8
Daub-5	93.5	93.0	94.2	97.3	97.0	98.0
Daub-10	93.5	90.2	94.2	97.3	94.8	98.0
Daub-15	94.3	87.0	94.0	97.5	92.5	98.0
Daub-20	95.2	82.8	94.2	97.8	89.0	98.7

Table 6.6: Average bit rate heuristic of the six test images. Figure 6.13 gives a more ‘readable’ plot of these numbers.

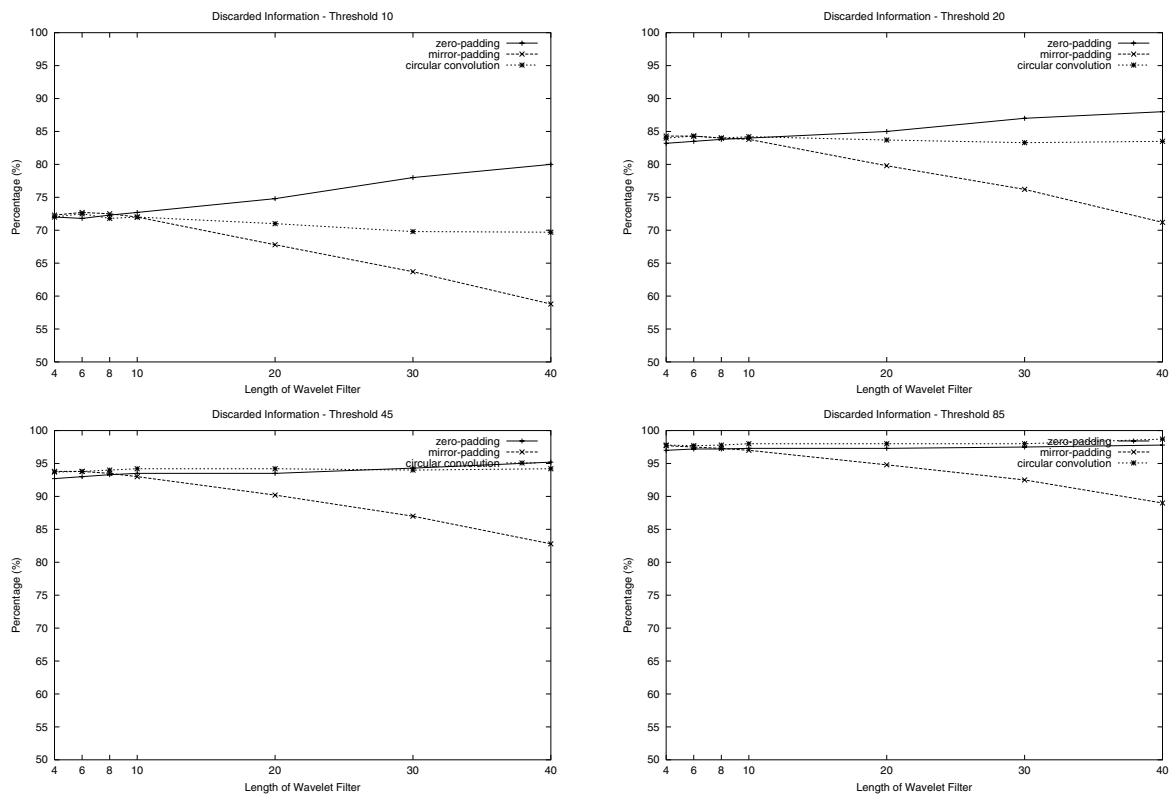


Figure 6.13: Average bit rate heuristic of the test images at the quantization thresholds $\lambda = 10, 20, 45, 85$. The values are averaged over the six test images and correspond to those in Table 6.6. Each plot covers a range of one dB for the PSNR.

Quality of visual perception — PSNR [dB]						
Wavelet	standard	non- standard	standard	non- standard	standard	non- standard
	<i>Mandrill</i>		<i>Brain</i>		<i>Lena</i>	
	Threshold $\lambda = 10$ — Excellent overall quality					
Daub-2	18.228	18.238	18.277	18.197	16.382	16.380
Daub-3	18.006	18.221	18.278	18.433	16.267	16.350
Daub-4	18.073	17.963	18.363	18.248	16.183	16.260
Daub-5	17.819	18.186	18.292	18.259	16.238	16.527
Daub-10	18.053	18.009	18.510	18.479	16.186	16.214
Daub-15	17.931	18.022	18.380	18.543	16.267	16.288
Daub-20	17.997	18.026	18.283	18.466	16.135	16.238
	Threshold $\lambda = 20$ — Good overall quality					
Daub-2	14.386	14.403	16.544	16.577	14.638	14.730
Daub-3	14.235	14.424	16.663	16.755	14.660	14.687
Daub-4	14.182	14.276	16.791	16.734	14.503	14.735
Daub-5	14.287	14.269	16.717	16.854	14.593	14.815
Daub-10	14.235	14.360	16.925	16.878	14.393	14.699
Daub-15	14.244	14.300	16.774	16.852	14.412	14.477
Daub-20	14.098	14.227	16.683	16.769	14.319	14.353
	Threshold $\lambda = 45$ — Medium overall quality					
Daub-2	10.832	10.910	14.832	14.747	12.895	12.832
Daub-3	10.895	10.948	15.191	15.052	12.846	12.903
Daub-4	10.842	10.885	15.053	15.056	12.713	12.818
Daub-5	10.901	10.949	15.103	14.999	12.919	12.983
Daub-10	10.889	10.913	15.047	15.212	12.684	12.795
Daub-15	10.805	10.815	15.019	15.064	12.623	12.686
Daub-20	10.756	10.843	14.895	14.882	12.609	12.640
	Threshold $\lambda = 85$ — Poor overall quality					
Daub-2	9.136	9.135	13.789	13.783	11.625	11.577
Daub-3	9.157	9.124	13.734	13.759	11.609	11.516
Daub-4	9.157	9.168	13.718	13.808	11.687	11.636
Daub-5	9.198	9.203	13.707	13.705	11.745	11.872
Daub-10	9.198	9.206	13.929	13.922	11.598	11.555
Daub-15	9.135	9.126	13.701	13.917	11.479	11.475
Daub-20	9.208	9.204	13.968	13.974	11.682	11.597
	<i>Camera</i>		<i>Goldhill</i>		<i>House</i>	
	Threshold $\lambda = 10$ — Excellent overall quality					
Daub-2	17.431	17.371	16.146	16.412	19.421	19.608
Daub-3	17.398	17.625	16.232	16.358	19.574	19.621
Daub-4	17.544	17.577	16.334	16.342	19.626	19.584
Daub-5	17.332	17.389	16.115	16.154	19.426	19.566
Daub-10	17.441	17.383	16.168	16.317	19.494	19.732
Daub-15	17.500	17.523	15.960	16.033	19.153	19.726
Daub-20	17.315	17.484	16.131	16.319	19.401	19.524
	Threshold $\lambda = 20$ — Good overall quality					
Daub-2	14.537	14.396	13.782	13.898	17.385	17.471
Daub-3	14.465	14.426	13.740	13.858	17.498	17.612
Daub-4	14.535	14.430	13.882	13.753	17.332	17.351
Daub-5	14.579	14.427	13.749	13.711	17.197	17.465
Daub-10	14.413	14.409	13.780	13.798	17.484	17.689
Daub-15	14.456	14.414	13.752	13.697	17.346	17.266
Daub-20	14.336	14.362	13.738	13.731	17.410	17.240
	Threshold $\lambda = 45$ — Medium overall quality					
Daub-2	12.209	12.131	12.001	11.876	15.167	15.155
Daub-3	12.225	12.188	12.086	11.889	15.114	15.118
Daub-4	12.222	12.145	11.990	11.925	14.928	15.180
Daub-5	12.226	12.120	12.038	11.801	15.168	15.244
Daub-10	12.199	12.093	11.988	11.726	15.183	14.872
Daub-15	12.118	12.176	11.914	11.753	14.790	14.969
Daub-20	12.148	12.048	11.981	11.703	15.233	14.609
	Threshold $\lambda = 85$ — Poor overall quality					
Daub-2	11.294	11.041	11.025	10.844	13.461	13.703
Daub-3	11.265	11.080	11.015	10.754	13.439	13.627
Daub-4	11.162	11.046	10.932	10.740	13.309	13.510
Daub-5	11.239	11.129	10.948	10.738	13.560	13.111
Daub-10	11.150	10.987	10.910	10.771	13.316	13.434
Daub-15	11.043	10.982	10.836	10.607	13.313	13.123
Daub-20	11.119	11.090	10.943	10.766	13.417	13.678

Table 6.7: Detailed results of the quality evaluation for the standard versus the nonstandard decomposition strategy. The mean values over the images are given in Table 6.8 and are visualized in Figure 6.14.

Average image quality — PSNR [dB]										
			standard	non-standard	standard	non-standard	standard	non-standard	standard	non-standard
Wavelet	taps	iterat.	Threshold $\lambda = 10$		Threshold $\lambda = 20$		Threshold $\lambda = 45$		Threshold $\lambda = 85$	
Daub-2	4	7	17.648	17.701	15.212	15.246	12.989	12.942	11.721	11.681
Daub-3	6	6	17.626	17.768	15.210	15.288	13.060	13.016	11.703	11.643
Daub-4	8	6	17.687	17.662	15.204	15.213	12.958	13.002	11.661	11.651
Daub-5	10	5	17.537	17.680	15.187	15.257	13.059	13.016	11.733	11.626
Daub-10	20	4	17.642	17.689	15.205	15.306	12.998	12.935	11.684	11.666
Daub-15	30	4	17.532	17.686	15.164	15.168	12.878	12.911	11.585	11.538
Daub-20	40	3	17.544	17.676	15.097	15.114	12.937	12.788	11.723	11.718

Table 6.8: Average quality of the six test images in the comparison of standard versus nonstandard decomposition. Figure 6.14 gives a more ‘readable’ plot of these numbers.

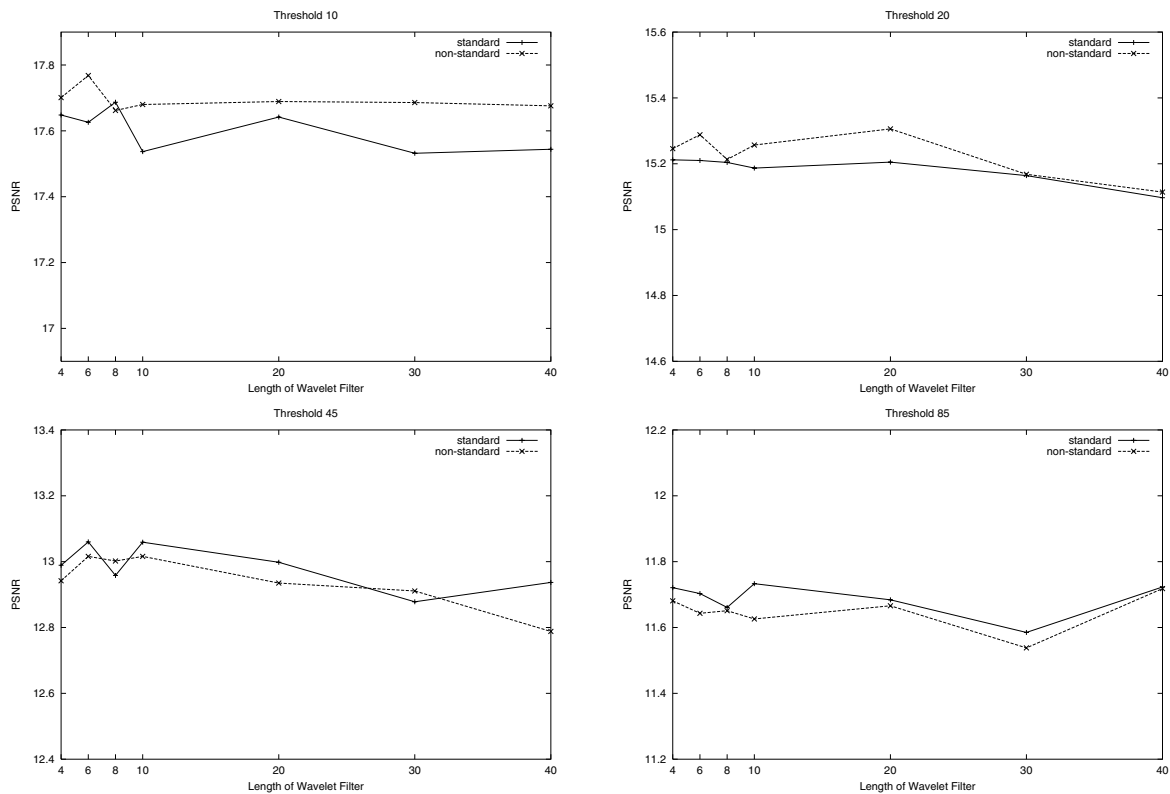


Figure 6.14: Mean visual quality of the test images at the quantization thresholds $\lambda = 10, 20, 45, 85$ with standard versus nonstandard decomposition. The values correspond to Table 6.8. Each plot covers a range of one dB for the PSNR.

6.4 Regions-of-interest Coding in JPEG2000

This section discusses a specific feature of the JPEG2000 coding standard which is based on the time-frequency (in this context: *location-frequency*) information of the wavelet transform on still images: coding of regions of specific interest within an image, the *regions-of-interest* (ROI). A brief overview of JPEG2000 precedes the presentation of different approaches for regions-of-interest encoding. In the master's thesis of Holger F   ler [F   01], elaborated at our department, we have demonstrated the strengths of region-of-interest coding. Its propagation, however, might depend on the integration of good segmentation algorithms into current image processing tools. We have included this topic in this book since the definition of a region-of-interest nevertheless represents a clever idea, and one which was non-existent in JPEG.

6.4.1 JPEG2000 — The Standard

The *Joint Photographic Experts Group* (JPEG) is a group of experts working on standards of image compression of both *International Standardizations Organization* (ISO) and *International Telecommunications Union* (ITU). In March 1997, they launched a call for participation in the development of the new image coding standard JPEG2000, which was declared a standard on January 2, 2001.

This section reviews the first part of JPEG2000, which defines the heart of the new standard. Further parts explain extensions in functionality, in motion-JPEG2000 coding, in conformance, and reference software, etc. Table 6.9 lists the different parts of the standard. We give a brief survey of the design and architecture of the standard [SCE00b] [ITU00].

Part	Content
1	JPEG2000 Image Coding System
2	Extensions
3	Motion-JPEG2000
4	Conformance
5	Reference Software
6	Compound Image File Format

Table 6.9: Structure of the JPEG2000 standard.

6.4.1.1 Design Goals

The design goals of the new standard which is meant to extend, not replace, the DCT-based JPEG standard can be summarized as follows [F   01]:

- *Better performance at low bit rate.* The performance of the new scheme shall be better than existing algorithms with regard to subjective and objective quality measures. Concerning the objective measure, the actual bit rate shall be close to the theoretical minimum at a given distortion.

- *Lossy and lossless compression.* A lossless modulus shall allow to archive, e.g., medical images which do not allow distortions.
- *Progressive data transmission.* Image transmission shall allow progressive refinement in both spatial resolution and pixel precision.
- *Definition and coding of regions-of-interest.* Specific regions of an image might be coded with higher precision than the rest (see Section 6.4.2).
- *Random access.* The data stream shall be coded such that specific parts of the image might be coded separately or in different order.
- *Robustness towards bit errors.* The coding of the data stream shall be robust towards transmission errors (e.g., in wireless networks), and the loss of data shall impact the smallest possible area of the image.
- *Open architecture.* The architecture shall allow a flexible adaptation to applications (e.g., efficient coding of images with specific properties).
- *Possibility of content description.* The data stream shall permit the integration of meta information to facilitate indexing.
- *Transparency.* The image might contain additional information about transparent regions since this is an important feature for the Internet.
- *Watermarking.* The data stream shall allow to incorporate information on the intellectual property rights to an image.
- *Support of images containing arbitrary components.* In contrast to JPEG, where images are restricted to a resolution of 3×8 bits per color, the new scheme shall allow more flexibility in color resolution.

6.4.1.2 Architecture

The JPEG2000 codec [ITU00] processes an image of arbitrary size and number of color components, each of them represented with arbitrary precision, according to the following steps:

1. The image is decomposed into its color components, which are processed separately.
2. Each color component is subject to a *tiling* process: The spatial representation is split into equal-sized, non-overlapping tiles of an arbitrary, but fixed size. Thereafter, each tile is treated separately.
3. Each tile is subject to the *wavelet transform*. It results in time-scale coefficients at different resolutions, see Chapter 1. The two standard filters for lossless and lossy coding have been presented in Section 3.6.
4. The different scales are ordered such that they describe specific regions of the image, i.e., the approximation of a *specific area* of the approximation is combined with coefficients of medium and fine resolution. Together, they describe the selected area of the image at high resolution. The resulting blocks of the ordering process are called *subbands*.

5. The subbands are quantized and stored in *code blocks*.
6. The bit layers of the coefficients in the code blocks are entropy-encoded.
7. A specific treatment of *regions-of-interest* is allowed which codes specific regions with greater precision than the rest.
8. The data stream is enriched by *markers* which allow recovery of transmission errors.
9. A header for the data stream is generated which contains the selected parameters and thus delivers to the decoder the information necessary to recover the image.
10. A file format allows the storage of the data stream, and a decoder synthesizes the images in the basis of the header information. It then knows which parts of the image shall be decoded in what order and with what precision.

The fundamental coding unit of the JPEG2000 standard is the *Embedded Block Coding Optimized Truncation* (EBCOT) algorithm described in [Tau00].

6.4.2 Regions-of-interest

The region-of-interest coding has evolved from the request to encode an image with maximum conservation of resources. Coding schemes like JPEG encode all parts of an image with equal quality: A JPEG-encoded image requires user interaction only to select the quantization threshold, thus for the overall coding quality of the image. This has the dual advantage of simplicity and therefore speed. A drawback of the JPEG approach is the fact that an image generally contains regions that are more important than others for the human visual perception.

If the encoding algorithm had information on visually important regions, it could utilize this knowledge through optimized compression [Füß01]. In the region-of-interest coding strategy, user-defined regions are marked which are coded with a higher quality than the background, i.e., the parts outside the region-of-interest. The JPEG2000 standard defines this option.

6.4.2.1 General Approach

In the general approach, a region-of-interest is a coding scheme concept that is further subdivided into two steps:

Step 1: Image Segmentation. The raw data of an image is segmented², and each segment is assigned a quality level. Algorithms for the segmentation include *manual*, *semiautomatic* (see also Section 6.2), *automatic* (e.g., with a face detector), and *constant* segmentation (i.e., a pre-defined and image-independent region, e.g., the central 10% of both the horizontal and the vertical image spread).

Step 2: Image Compression. The segmented and quality-level-enriched raw data is transferred to a compression algorithm, which employs the available information. The encoder could split the original

²The *shape* of the segmentation is discussed in Section 6.4.2.3

signal according to the assigned segments of equal quality and then encode each quality level with an algorithm like JPEG with different quantization thresholds. In fact, JPEG2000 pursues a different approach, as we will see later.

The output of the second step is an encoded data stream which contains additional information on visually important regions, in contrast to that delivered by a standard hybrid encoder.

6.4.2.2 What Is of Interest?

The investigation of a region-of-interest requires a pragmatic approach to the term ‘interest’. Since the semantic interpretation of ‘interest’ does not suffice in this context, the two notions of *regions of higher coding quality* (RHQ) and *regions of minor quality* (RMQ) are introduced. We further distinguish between two classifications of segmentation.

Classification according to information content. Different parts of an image generally contain different information content. In portraits (of a human, an animal, a machine, etc.), the portrayed subject clearly carries most of the information content of an image. Thus an obvious approach is to define the background of a portrait as an RMQ and the portrait itself as an RHQ. Figure 6.15 demonstrates this classification by means of the image *Lena* on the left and a bitmask for the region-of-interest on the right. The RHQ is marked in black, while the background, i.e., the RMQ, is painted white.



Figure 6.15: Classification according to image content.

Classification according to visual perception. The human visual system is more sensitive to distortions in the foreground of an image than to any in the background. A classification according to the ‘remoteness’ of visual objects, therefore, is a second possible approach (see Figure 6.16). Another option is the segmentation according to uniformity, where textured areas are visually more important (RHQ) than uniform areas (RMQ).



Figure 6.16: Classification according to perception of distance.

6.4.2.3 Shape of Region-of-interest Segments

The segments selected for the coding of a region-of-interest might be of arbitrary shape according to the image under consideration (see Figures 6.15 and 6.16 as examples) or of a specific pre-defined shape and/or size (i.e., constant segmentation). The precise definition of the shape is generally a trade-off between efficient data storage and flexibility. Figure 6.17 gives two examples of pre-defined regions-of-interest, marked in white.



Figure 6.17: Two examples of a pre-defined shape of a region-of-interest.

For simple shapes of a region-of-interest, methods have to be implemented which generate the corresponding bitmask. Arbitrary shapes complicate description in the data stream and might negatively influence the total compression ratio. An automatic segmentation algorithm for the definition of an arbitrary region-of-interest mask might prove useful for specific classes of images, when prior knowledge of the images can successfully be exploited to implement a robust segmentation approach (see Section 6.2.1). Examples of such image classes include *identity photos* or *finger prints*.

6.4.2.4 Number of Region-of-interest Segments

Until now, we have discussed the special case of a yes/no-bitmask for the definition of a region-of-interest, i.e., a specific pixel of an image can either be of interest (RHQ) or not (RMQ). In a general approach, an arbitrary number of region-of-interest segments could be assigned to an image, where each segment defines a specific quality level. If a pixel is assigned to more than one region-of-interest, it would be coded for the region with the highest assigned quality, thus avoiding multiple coding. Again, a trade-off has to be found between the complexity and the flexibility of the algorithm. Figure 6.18 shows an example of a region-of-interest mask with three quality levels, where the assignment to a level was made according to the remoteness of the area.



Figure 6.18: Region-of-interest mask with three quality levels: black = important; gray = medium; white = less important.

6.4.2.5 Regions-of-interest in JPEG2000

Part 1 of JPEG2000 allows regions-of-interest of arbitrary shape, but of only one quality level, i.e., a yes/no-bitmask. The encoding of a region-of-interest is based on the *MAXSHIFT* method of [ITU00, Annex H]. A description of this algorithm can be found in [CAL00] and [Füß01]. The core idea of the MAXSHIFT method is to change the *relative* position of the bit levels between RHQ and RMQ so that the coefficients of a region-of-interest are assigned to higher bit levels than the background coefficients. This enlarges the required storage space; the upper limit is a doubling of the bit levels in the event that the highest bit of a background coefficient is occupied. In the subsequent data transmission, the information is encoded according to the bit layers. Quantization enters if the data stream is interrupted or the decoder decides not to receive the complete information. In any case, this scheme assures that the coefficients within a region-of-interest are transmitted prior to the other coefficients of the same code block.

The importance of the MAXSHIFT method is due to the fact that it manages to encode a region-of-interest *implicitly* in the data stream; transmission of the shape of the region-of-interest is not necessary. The decoder needs information only on the amount s of bit shifts to increment the bit levels of those coefficients accordingly, whose highest bit level is smaller than s .

6.4.3 Qualitative Remarks

In the original meaning of the algorithm, the MAXSHIFT method was intended to ensure that the most important information of an image was transferred prior to the less important parts. It was not intended to code different quality levels in a fully transmitted image. But in practice, the precision of the coefficients outside a region-of-interest actually decreases since bit levels are *truncated*.

A region-of-interest is defined per tile and per code block. Thus, various combinatory definitions of a region-of-interest result. Examples are definitions of a region-of-interest according to different color components, or according to different tiles.

Part 2 of the JPEG2000 standard contains a number of extension features, including a generalization of the notion of region-of-interest [ITU00, Annex K]. In contrast to the possibilities of the MAXSHIFT approach, the most important generalizations are:

- Definition of an *arbitrary number* of regions-of-interest, each of which is assigned a parameter s for bit scaling, thus resulting in arbitrary quality levels.
- In addition to arbitrary shapes, ellipsoids are pre-defined by means of their center point, width, and height.

At the University of Mannheim, we have implemented parts of JPEG2000's region-of-interest scheme in the context of the master's thesis of Holger Fußler [Füß01]:

- tiling,
- wavelet transform with either the Daub-5/3 or the Daub-9/7 filter (see Section 3.6),
- generation of a region-of-interest mask, and
- application of the MAXSHIFT method with subsequent 'quantization'.

Though region-of-interest coding makes use of the attractive property to distinguish between regions of different importance during the encoding process, its success in practical usage will depend on the ease of *defining* the visually important regions of an image. Therefore, we expect pre-defined regions-of-interest like those in Figure 6.17 to constitute the overwhelming majority of region-of-interest shapes. Furthermore, a digital image library with several 100 Mbyte of data transfer per day might use the region-of-interest scheme to reduce network load. This reduction, however, might be less a reduction of storage space, but the feature could be used for hierarchical data transmission. In this scenario, an application would not only request an image, but an image of specific quality. Region-of-interest-encoded images could then allow to scale background coefficients first.

The latter scenario plays into the discussion of a scalable image server which has to meet time specifications. Since video data is strongly time dependent these limitations are even more relevant for real-time video servers. The scalability of digital image and video data is addressed in the following chapter.

Chapter 7

Hierarchical Video Coding

In research the horizon recedes as we advance, and is no nearer at sixty than it was at twenty. As the power of endurance weakens with age, the urgency of pursuit grows more intense. . . And research is always incomplete.
– Mark Pattison

7.1 Introduction

We have seen that the wavelet transform finds promising applications in audio content analysis, where a one-dimensional signal over time is being analyzed and denoised. On still images, the wavelet transform provides the basis for the best-performing compression algorithms that are known so far; it has especially entered into the JPEG2000 standard. One of the questions researched during this dissertation was to what extent the wavelet transform could be successfully exploited in *hierarchical* video coding. Preliminary results were presented at [KKSH01], [KS01], and [SKE01a].

Streaming video is regarded as one of the most promising Internet applications of the future. Nevertheless, a major drawback to its rapid deployment is the heterogeneity of the Internet. The available bandwidth is a major parameter for the quality of real-time streaming applications: The more bandwidth there is available, the better the quality of the video can be. But available bandwidth varies from user to user.

In the teleteaching project VIROR [VIR01] of the University of Mannheim, within which much of the presented work was carried out, the cooperating Universities of Freiburg, Karlsruhe, Heidelberg, and Mannheim are connected with a high-speed ATM network of 155 Mbit/s with a guaranteed bandwidth of 10 Mbit/s. This allows us to transmit high-quality video of up to 3 Mbit/s between the participants.

For private access, the situation is different. Broadband Internet connections are still costly, thus the majority of private users in Germany still connect to the Internet via an analog modem or ISDN. A modem allows an average data rate of 30 to 50 kbit/s, which is merely sufficient to receive audio in an acceptable quality. With ISDN access, the data rate is 64 or 128 kbit/s. An access bandwidth of 128

kbit/s permits the reception of audio and video in a still poor, but sufficient quality to be able to follow the contents. Recently, Deutsche Telekom has been promoting the *Asynchronous Digital Subscriber Line* (ADSL) technology. Technically, ADSL allows a downstream of 6 to 9 Mbit/s but Deutsche Telekom offers only 768 kbit/s to private customers.

Consequently, an encoded video stream should be scalable for different network capacities. This is accomplished through *layered* or *hierarchical* video streaming. The subdivision of an encoded data stream into different layers enables the user to receive (in the ideal case) exactly as much data as his/her individual facilities allow. Figure 7.1 shows an example.

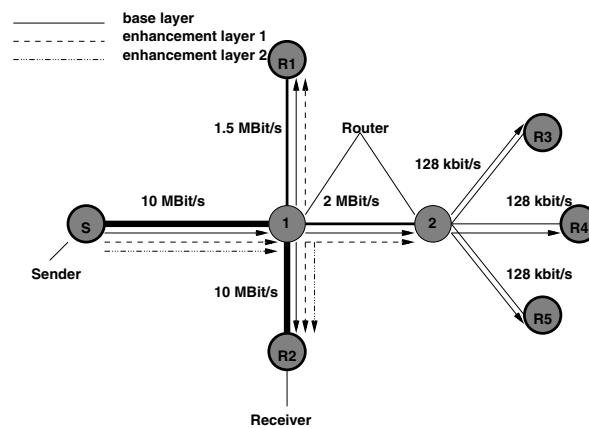


Figure 7.1: Layered data transmission in a heterogeneous network. The sender sends the base layer plus all enhancement layers. Each receiver chooses how many layers he/she can receive according to the bandwidth available.

The goal of good video scalability led us to search for a good layering technique. Advantages and drawbacks differ with regard to practical aspects. Various layering algorithms which are relatively easy to integrate into current coding standards like MPEG are based on the discrete cosine transform (DCT). New algorithms, however, focus on the discrete wavelet transform since the mixture of time, respectively, location and scale information in the wavelet-transformed space can be successfully exploited to provide better quality at lower bit rates. Moreover, the wavelet transform is of complexity $O(n)$ in contrast to the complexity $O(n \log n)$ of the DCT.

7.2 Video Scaling Techniques

Video can be interpreted as a vector consisting of three measurements: color resolution, spatial resolution, and temporal resolution. The *color resolution* is defined by the number of bits for the color value of each pixel. The *spatial resolution* describes the horizontal and vertical stretches of each frame. The *temporal resolution* describes the number of frames per second [Poy96]. Formally, a video can be defined as follows:

Definition 7.1 A color video V consists of a sequence of frames

$$V = \{(F_0, F_1, F_2, \dots)\},$$

where each frame F_k is composed of a number of pixels:

$$F_k = \{x_{ij} = (Y, U, V) \in [0, 255]^3 \mid i=0, \dots, w-1, j=0, \dots, h-1\}.$$

Here, w denotes the width, and h denotes the height of the frame sequence. The triple $(Y, U, V) \in [0, 255]^3$ defines the luminance and the two chrominance components of a color.

Hierarchical encoding techniques scale the video quality in at least one of the above three resolutions. The idea is to encode video signals not only into one but into several output streams: a base layer l_0 and one or several enhancement layers l_i ($1 \leq i$). Each layer l_i depends on all lower layers l_0, \dots, l_{i-1} , it can only be decoded together with these lower layers, each of which adds to the quality of the video. In the following, we give a generalized definition of [McC96] for a hierarchical encoder and decoder [KKSH01].

Definition 7.2 Let $V_{i,k}$ be a sub-sequence of length k of the video V , starting at frame $i + 1$:

$$V_{i,k} = (F_{i+1}, \dots, F_{i+k}).$$

A hierarchical encoder E encodes a sequence of k frames into L output codes C^1, \dots, C^L . Therefore, E is a mapping

$$E : V_{i,k} \rightarrow \{C_{i,k}^1, \dots, C_{i,k}^L\}.$$

In order to reassemble the video at the receiver side we need a decoder D that reverses codes $C_{i,k}^1, \dots, C_{i,k}^l$ into a sequence of frames:

$$D : \{C_{i,k}^1, \dots, C_{i,k}^l\} \rightarrow (\hat{F}_{i+1}, \dots, \hat{F}_{i+k}) = \hat{V}_{i,k}, \quad l \leq L.$$

The difference between the original sub-sequence $V_{i,k}$ and the decoded sequence $\hat{V}_{i,k}$ shortens with the number of codes l taken into account at this inversion.

According to Definition 7.2, the elementary task of a hierarchical encoder E is to define encoding schemes that split (and compress) a given frame sequence into a set of codes $\{C^l\}$.

A number of hierarchical video coding techniques have been developed to scale and compress a frame sequence in its three resolutions: time, size, and color depth. Color scaling was beyond the scope of this dissertation. In the following, we briefly summarize the most common approaches to temporal and spatial scaling. A more detailed overview can be found in [KK98].

7.2.1 Temporal Scaling

Temporal scaling approaches are quite intuitive: They distribute consecutive frames of a video sequence over a number of different layers. Figure 7.2 visualizes a possible approach with three layers, where a subsample of the image sequence is transmitted on each layer [MFSW97]. In other words, the more layers are received, the higher the *frame rate* will be.

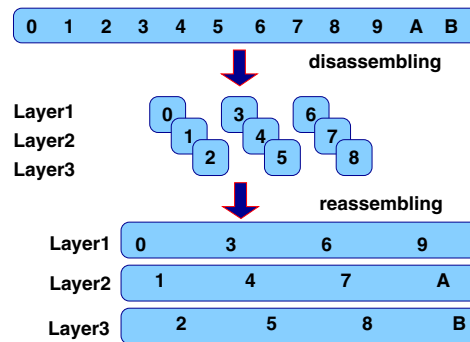


Figure 7.2: Temporal scaling of a video stream.

7.2.2 Spatial Scaling

The majority of spatial scaling approaches splits each video frame into its spatial frequencies: Implementations either produce a set of spatially smaller copies of a video, or they scale the coefficients obtained by a transformation into the frequency domain. Since lower spatial frequencies are better perceived by human observers [RF85], the lower layers of spatial scaling approaches contain the lower frequencies, while higher layers provide information about higher spatial frequencies.

At the University of Mannheim, Christoph Kuhmünch implemented the following common spatial scaling schemes in the context of his dissertation [Kuh01].

- *Pyramid Encoding.* The central idea of this approach [BA83] is that the encoder first downsamples the image, compresses it according to the chosen encoding technique, and then transmits it in the base layer stream. When the image is decompressed and upsampled, a much coarser copy of the original arises. To compensate for the difference, the encoder subtracts the resulting copy from the original image and sends the encoded differential picture in the enhancement layer stream. This approach is used in the MPEG-2 video standard [ISO95].
- *Layered Frequencies.* In this approach, each 8×8 block of each frame of a digital video is transformed into the frequency domain using the *discrete cosine transform* (DCT), see Section 9.4 for its definition. After quantization, the coefficients are stored in different layers [McC96]. For instance, the base layer contains the first three coefficients of the transformed block, the first enhancement layer contains the next three coefficients, etc.
- *Layered Quantization.* In [PM93] [AMV96] [McC96], a spatial approach is described which relies on layered quantization of a DCT-encoded frame sequence: Each 8×8 block of each image

is transformed into the frequency domain. The bits of the DCT coefficients are distributed over several layers. This corresponds to applying different quantization factors to the coefficients, ranging from coarse to fine.

Clearly, the visual quality of the video received depends on the construction of the different layers at the encoder. Keeping in mind that the data sink for video transmission over the Internet is a human observer, the task is to find an algorithm that maximizes the perceived quality of the video. Here we enter the domain of video quality metrics.

7.3 Quality Metrics for Video

The optimization of digital image processing systems with respect to the capture, storage, transmission, and display of visual information is one of the major challenges in image and video coding. The consideration of how people perceive visual information proves to be very useful in this field. For instance, quality assessment tools predict subjective ratings, and image compression schemes reduce the visibility of introduced artifacts.

7.3.1 Vision Models

The modeling of human perception of visual stimuli is a field of ongoing research. While the *human visual system* (HVS) is extremely complex and many of its properties are not well understood even today, models of human vision are the foundation for accurate and general metrics of visual quality. One of the first books to present a broad overview of how human observers see is [Fri79]. More recent research is presented in [Wan95], where the human visual system is explained with respect to the MPEG compression standard. Psycho-physiological research has been carried out in order to measure the sensitivity of the human visual system in the three domains of color, spatial, and temporal resolution. These research projects proved the following attributes of the human visual system:

1. Human visual perception is based less on absolute (luminance) values and more on contrast [vdB96].
2. Contrast sensitivity is much higher for luminance than for chrominance [Mul85].
3. Contrast sensitivity is highly correlated to the spatial frequency of the perceived stimulus and decreases if spatial frequency increases [vNB67].
4. The critical flicker frequency, i.e., the minimum number of frames per time unit that make a video appear ‘fluid’, is highly correlated to luminance and motion energy [MPFL97].

Based on these results, a number of mathematical models have been designed that simulate the human visual system. These models finally lead to the proposal of quality metrics.

For still images, we refer to the following two models: (1) Modeling of the human visual system by imprecise data sets is presented in [Ste98]. (2) A model based on the wavelet transform is presented in

[Boc98]. Each scale within the wavelet-transformed domain is accredited to a specific weight which was found empirically. According to this weight, distortions have different influences on the visual perception. Other metrics for still images have been proposed in [ICB01] [SHH01] [FTWY01].

An overview of vision models for the perception of video can be found in [Win00]. Furthermore, an isotropic measure of local contrast which is based on the combination of directional analytic filters is proposed in [Win00].

7.3.2 Video Metrics

The advent of digital imaging systems has exposed the limitations of the techniques traditionally used for quality assessment and control. Due to compression, digital imaging systems exhibit artifacts that are fundamentally different from those of analog systems. The amount and visibility of these distortions depend on the actual image content. Therefore, traditional measurements are inadequate for the evaluation of these artifacts. Since the subjective assessment of video is tied up with time-consuming and expensive tests, researchers have often sought suitable metrics for an algorithmic approximation of the human visual perception.

7.3.2.1 The ITS Metric

A first attempt to widen the models of human visual perception into the spatio-temporal dimension, and thus to adapt them to digital videos, is the *ITS metric* of the Institute for Telecommunication Sciences presented in [WJP⁺93]. The visual quality measure proposed in this work relies upon two quantities. The first one measures spatial distortions by comparing edge-enhanced copies of the original to their corresponding approximation frames. The latter measures the loss of temporal information by comparing the motion energy of the original with that of the approximation frame sequences. These two units of information are post-processed by three measures whose weighted linear combination conforms with the results of subjective testing, a scale ranging from 1 (i.e., very poor quality) to 5 (i.e., excellent quality).

7.3.2.2 The DIST Metric

A sophisticated distortion metric is proposed in [vdB96]. It relies on the two basic assumptions that a human observer does not perceive an image at the pixel scale nor does he/she ever see the whole image at a given instant. The distortion metric *DIST* therefore works on three-dimensional blocks of the video sequence: x -axis and y -axis for spatial dimension, and t -axis for temporal dimension. A subsequent division into different channels allows different fine-grained metrics. As the exact construction in [vdB96] remains unclear, we have implemented the *DIST* metric as follows:

$$\text{DIST} = \left(\frac{1}{N_c} \sum_{c=1}^{N_c} \left(\frac{1}{N_x N_y N_t} \sum_{t=1}^{N_t} \sum_{y=1}^{N_y} \sum_{x=1}^{N_x} |e[x, y, t, c]| \right)^\beta \right)^{\frac{1}{\beta}},$$

where $e[x, y, t, c]$ is the error between the distorted frame and the original at time t and position (x, y) in channel c , N_t , N_x , and N_y are the dimensions of the block, and N_c is the number of channels, where a channel is a bandpass-filtered (i.e., downsampled) image. The parameter β was set to 4.

7.3.2.3 The Peak Signal-to-noise Ratio

Looking for fast alternatives to the above video quality metrics, we have turned to the *peak signal-to-noise ratio*:

$$\text{PSNR [dB]} = 10 \cdot \log \left(\frac{\sum_{xy} 255^2}{\sum_{xy} (F_j(x, y) - \hat{F}_j(x, y))^2} \right),$$

where $F_j(x, y)$ depicts the pixel value of the original frame F_j at position (x, y) , and $\hat{F}_j(x, y)$ denotes the pixel value of the decoded frame \hat{F}_j at position (x, y) . The value of 255 in the numerator depicts the maximum possible difference between the original and the decoded frame for grayscale images coded with a precision of one byte per pixel.

The PSNR as defined above gives an objective measure of the difference between single frames of a video sequence. In order to consider the time component, the PSNR is averaged over a number of consecutive frames (usually 25 frames), or over the overall time spread of a video sequence.

This physical measure operates solely on a pixel-by-pixel basis by comparing the values of the difference between subsequent frames. In other words, it neglects the actual image content and the viewing conditions. Nevertheless, we will show that the PSNR provides results comparable to those of the above metrics.

7.4 Empirical Evaluation of Hierarchical Video Coding Schemes

In this section, we present our empirical results on the performance of four different hierarchical video coding algorithms. Three of the presented algorithms were based on the discrete cosine transform, while the fourth algorithm was wavelet-based. These coding schemes were subjectively rated in a field trial where 30 test persons were asked to judge the quality of several test videos coded with the different schemes. The quality of the encoded videos was also computed with the objective video quality metrics introduced above. The correlation between the subjective ratings and the outcome of the metrics served as the performance indicator for the metrics. The results were presented in [KS01].

7.4.1 Implementation

The implementation was carried out within the scope of the master's thesis of Uwe Bosecker [Bos00] at our department in Mannheim. The operating system was Linux with a S.u.S.e. distribution. The programming language was C++.

Four different spatial video scaling algorithms were used: The algorithms *A1* to *A3* are based on the discrete cosine transform, and part of the implementation of [Kuh01] was re-used. The algorithm *A4* implements the discrete wavelet transform for the purpose of comparing both transforms.

- A1: Pyramid encoding.* The base layer contains a downsampled version of the original. Each enhancement layer contains the difference between the original and the upsampled – thus blurry – lower layer.
- A2: Layered DCT frequencies.* The base layer contains the *DC* (i.e., direct current) coefficients of each DCT block. The enhancement layers subsequently contain the *AC* (i.e., alternating current) coefficients in decreasing order of importance.
- A3: Bit layering or Layered DCT quantization.* Again, the base layer contains the *DC* coefficients. The enhancement layers contain the most significant bits of the *AC* coefficients at each level.
- A4: Layered wavelet-transformed coefficients.* The coefficients in the time-scale domain of the wavelet transform are ordered according to their absolute value and then stored in the different layers.

Since our evaluations on the quality of wavelet-encoded still images proposed to use filter banks of medium length, we decided to implement a (separable) Daub-6 filter bank. The iteration was selected to be nonstandard decomposition, and the boundary treatment was set to circular convolution. Thus, the number of iterations on the approximation part of the wavelet-transformed video depended on the spatial size of the frames.

Three different video quality metrics (see Section 7.3.2) were implemented to automatically rate codec quality at different levels: the ITS and the DIST metrics, and the peak-signal-to-noise ratio. In our tests it turned out that the ITS metric varied too little for our applications: Indeed the output varied only in the range of 4.55 to 4.77 on a scale of 1 to 5, no matter what the distorted video looked like. We thus restricted the evaluation to the DIST metric and to the PSNR.

7.4.2 Experimental Setup

The subjective evaluation of the metrics and the coding algorithms was carried out on several short video sequences of about 15 seconds length in *common intermediate format* (CIF) with 25 frames per second. In total we evaluated seven video sequences of different types, including animations as well as natural video with little and much action, scenes with subtitles, and b/w movies.

Two shots were taken from movie intros: *Mainzelmännchen* and *Warner Bros.* The first intro is a cartoon; it was chosen for its sharp contrast with clear edges, and the second because it displays text of varying contrast. We further tested the b/w video *Laurel and Hardy* with subtitles. Two shots were taken from the movie *The Matrix*. While the first shot featured high motion values (i.e., two people fighting), the second one contained very little motion. Another film was the home video *Schloß Mannheim*, which shows a person walking slowly at the beginning and later starting to run. This video thus produced low and high motion values. Since the home video was recorded with a mid-priced digital camera, it showed slight distortions usually not present in professional movies: a very

soft flickering of the luminance, probably produced by the auto focus of the camera. Finally, we took a shot from the comic strip *Werner*.

Thirty test persons rated the perceptual quality on a scale from 1 (excellent quality) to 5 (poor quality). All videos were coded with the four algorithms A1 to A4. The quantization parameters of the algorithms were varied in such a way that they all produced different levels of quality.

The probands' descriptive statistics are as follows: 55% male, aged 16 – 68 with an average age of 32.6 years. Half of the probands ranked themselves as having an average knowledge of digital video, 23% had no previous knowledge, and 27% claimed to have a good knowledge of digital videos.

Two hypotheses $H_{1;0}$ and $H_{2;0}$ were tested in our setup [KS01]:

$H_{1;0}$: The video metric DIST correlates better with the human visual perception of video than does the much simpler PSNR.

$H_{1;1}$: The video metric DIST does not correlate better with the human visual perception of video than does the much simpler PSNR.

$H_{2;0}$: The four layered coding schemes produce comparable video quality at a given bit rate.

$H_{2;1}$: The four layered coding schemes produce different video quality at a given bit rate.

7.4.2.1 Hypothesis $H_{1;0}$

The test scenario for the evaluation of the correlation between the *human visual perception* (HVP) and the automatic assessment was set up as follows: All video sequences were coded with all four algorithms A1 to A4 in such a way that the PSNR of each video sequence was comparable for the four coding schemes. To obtain different qualities, the PSNR was varied to different quality levels for two of the videos. The resulting eleven sequences are specified in Table 7.1. Figure 7.3 shows a frame of the test sequence *Mainzelmännchen*, coded with the four different algorithms. These screenshots were taken at different qualities, but they visualize the principal perceptual differences of the artifacts of each of the four hierarchical video algorithms.

7.4.2.2 Hypothesis $H_{2;0}$

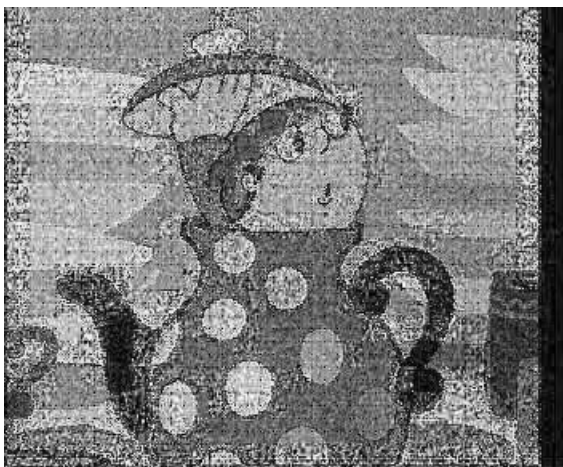
The parameters of the second scenario were set such that the videos produced (almost) the same bit rate, in contrast to the setup of $H_{1;0}$. As the discrete cosine transform is used in many standards, much research has been carried out in order to optimize its bit stream. For example, powerful entropy encoding schemes exist. In order to gain a fair estimate of the bandwidth requirements of each coding scheme, we restricted the entropy encoding to the same simple Huffman encoding scheme. The probands were shown the original video plus four distorted ones, coded with the schemes A1 to A4, at the same bit rate. As all coding schemes produce different artifacts at low bit rates, the question was which of them would be perceived as the visually least disturbing.



(a) A1: Pyramid encoding.



(b) A2: Layered DCT frequencies.



(c) A3: Bit layering.



(d) A4: Layered wavelet-transformed coefficients.

Figure 7.3: Visual aspect of the artifacts of different hierarchical coding schemes with the test sequence *Mainzelmännchen*.

Video Sequence	subject. rating	DIST	PSNR [dB]
Mainzelmännchen	4.50	2.63	64.7
Warner Bros.	1.77	0.83	73.4
Laurel & Hardy 1	3.57	2.67	59.9
Laurel & Hardy 2	4.70	3.50	56.1
The Matrix 1	2.30	0.67	76.8
The Matrix 1	4.90	2.83	63.4
Schloß Mannheim	2.73	3.13	63.1
Werner 1	1.10	2.23	68.4
Werner 2	1.90	3.10	61.7
Werner 3	4.30	4.87	53.1
Werner 4	4.87	5.67	50.0

Table 7.1: Test sequences for hypothesis $H_{1,0}$ and the results of the probands' average subjective ratings, the DIST metric and the PSNR.

7.4.3 Results

7.4.3.1 Hypothesis $H_{1,0}$: The video metric DIST correlates better with the human visual perception of video than does the much simpler PSNR

The results of the statistical analysis of correlation between the subjective quality ratings and the metrics are given in Table 7.2 [Bos00]. The hypothesis $H_{1,0}$ was answered by calculation of the Pearson correlation coefficient between the subjective ratings by the probands and the automatic assessments of the metrics. All difference hypotheses have been verified by variance analysis, where t-tests allowed the comparison to pairs of average mean. To verify the influence of socio-demographic characteristics (i.e., age, sex) and covariates (i.e., expertise), t-tests, and variance analyses were carried out with post-hoc tests [Kuh01].

	PSNR [dB]	DIST	DIST _Y	DIST _U	DIST _V
A1	-0.89	-0.69	-0.81	-0.62	-0.64
A2	-0.68	0.41	-0.66	0.25	0.34
A3	-0.71	-0.68	-0.73	-0.61	-0.71
A4	-0.70	-0.66	-0.79	-0.62	-0.58

Table 7.2: Correlation between the human visual perception and the PSNR, respectively the DIST metric and its sub-parts. Significant correlations are highlighted.

For a better understanding of the correlation between the human visual perception and the objective video quality metrics, we present not only the accumulated metric DIST, but also its sub-metrics of the luminance channel DIST_Y, and the two chrominance channels DIST_U and DIST_V. Table 7.2 clearly states that there is a correlation between the human visual perception of video and both metrics: PSNR and DIST. As the PSNR is constructed such that lower values denote lower quality, but the subjective rating was performed on a scale where lower values denote better quality, the sign for the

PSNR is negative. The same holds true for the DIST metric. The two chrominance parts of the DIST metric, $DIST_U$ and $DIST_V$ are much less correlated to HVP, and thus worsen the total result for the DIST metric. Regarding the absolute values of correlation between a video quality rating and the video metrics in the test, we see that the PSNR wins the bid for all four coding schemes $A1$ to $A4$. Sole consideration of the luminance value $DIST_Y$ though reveals results very close to the correlation between the PSNR and the human visual perception.

In the overall evaluation, we state that the DIST metric does not reach the performance level of the PSNR, but that both PSNR and the luminance sub-component $DIST_Y$ show a significant correlation to the subjective assessment of video quality. Neither maps perfectly to the subjective ratings by the test probands, but with less than 1% error probability, the mapping is sufficiently good. We had stated that the DIST metric claims to mirror human perception and that its computing complexity is rather high. The PSNR, by contrast, is very simple. The above results, however, do not justify any superiority of the DIST or the $DIST_Y$ metric towards the PSNR.

7.4.3.2 Hypothesis $H_{2,0}$: The four layered coding schemes produce comparable video quality at a given bit rate

Our evaluation of a best-performing hierarchical video encoder of our four implemented algorithms $A1$ to $A4$ is restricted to the quality assessment. It was measured by means of the subjective rating as well as with the PSNR and the DIST metrics. The results are presented in Table 7.3 [Bos00].

	Subjective Rating				PSNR [dB]	DIST
	average	variance	min	max		
$A1$	3.39	0.42	2.71	4.29	58.43	4.48
$A2$	3.54	0.43	2.71	4.29	58.34	4.08
$A3$	4.20	0.48	3.14	5.00	53.29	13.64
$A4$	2.98	0.49	2.00	4.14	63.26	3.62

Table 7.3: Evaluation of the four layered video coding schemes.

Note that the range of the rating encompasses the scale from 1 (excellent) to 5 (poor). The values for a given algorithm are averaged over all video sequences. Since the rating improves with decreasing values, the discrete wavelet transform ($A4$) wins the competition, followed by pyramid encoding ($A1$), layered DCT ($A2$), and bit layering ($A3$). A variance analysis accounts for the statistical relevance of the algorithms' effects on human visual perception.

7.4.4 Conclusion

The empirical tests have particularly proven that, despite its complexity, the DIST metric shows poor correlation to human visual perception. Its luminance sub-part $DIST_Y$, however, reveals a correlation to human visual perception comparable to that of the PSNR, so that both metrics mirror the quality as assessed by the subjective ratings of test persons. Nonetheless, as the results of $DIST_Y$ are not

convincingly better than the output of the PSNR, we conclude that it is not worth the cost of implementation and use.

The second test setup showed that the wavelet transform produces the best subjective quality at a given bandwidth. Admittedly, the results of all coding schemes are close. However, an implementation with more sophisticated entropy encoding schemes could reveal further performance differences.

The rather disappointing experience with sophisticated video quality metrics led us to concentrate on the PSNR for the evaluations presented in the following sections.

7.5 Layered Wavelet Coding Policies

In Chapter 3 we have discussed the numerous parameters of a wavelet-encoded video: choice of the wavelet filter bank, decomposition depth of the analysis (i.e., number of iterations on the low-pass filtered part), and decomposition type (i.e., standard or nonstandard). In Section 7.2 we have introduced different layering policies for hierarchical coding. Obviously, the most important information of a video has to be stored in the base layer l_0 , while less important information has to be stored stepwise in the enhancement layers l_i . However, the *ranking* of the importance of the information depends on the layering policy.

Inspired by the positive results of the evaluation on the quality of a wavelet-based layered video codec in the previous section, this section expands the research on layering policies for wavelet-based video encoders. We present and discuss three different wavelet-based layered encoding schemes. Based on the parameters information rate, bit rate, scalability, and human visual perception, we develop a recommendation according to which the different units of information should be distributed over the different layers of the video stream to result in maximal perceived quality. In this context, we focus again on spatially scaled video. The video sequence is thus regarded as a sequence of still images, and the wavelet transform is performed on each single frame. Our discussion of the different wavelet-based hierarchical video encoding schemes and their empirical evaluation was presented in [SKE01a].

7.5.1 Layering Policies

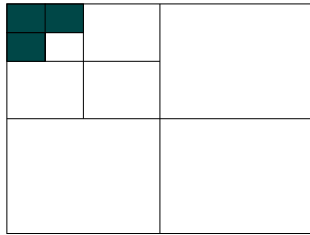
When an image is presented to a person, he/she first resolves the greater context of the situation: a car, a donkey, a crowd of people. Subsequently, more and more details enter the perception: the model and color of the car, the individuals in the crowd. Finally, details might be resolved: scratches in the varnish, expression of joy on a face [Fri79]. This property of human perception: working from coarse scale to fine scale, is reflected in the multiscale analysis of the wavelet transform. Reasonable layering of wavelet-transformed data can be carried out according to the three policies demonstrated in Figure 7.4 [SKE01a].

Policy 1: Blockwise. In Section 1.6 we mentioned that the low scales in a multiscale analysis best approximate the frequencies that are most important for human visual perception. Consequently, the layering and its respective synthesis work just the other way around: The low-pass filtered parts are

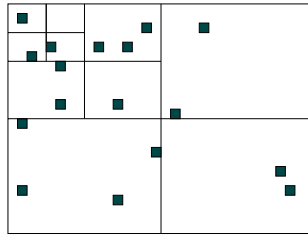
synthesized first, and if there is still a capacity for further synthesis, the high-pass filtered blocks are successively included in the decoding process. This is illustrated in Figure 7.4 (a).

Policy 2: Maximum coefficients. In contrast to the argument above, one could claim that the philosophy of wavelet decomposition is to concentrate the energy of a signal (and thus the information most important to human perception of video) in those coefficients in the time-scale mixture of the wavelet domain that have the highest absolute value, no matter where in the wavelet-transformed domain these coefficients are located. Consequently, the first iteration l_0 of the layering process should look for those coefficients with the highest (absolute) values, i.e., those above a certain threshold n_0 . Subsequent layers l_i are filled with the coefficients of the time-scale domain above smaller thresholds $n_i < n_{i-1}$ for $i \geq 1$, but still below the higher threshold n_{i-1} set before. In other words, each layer is filled with difference data at decreasing thresholds. This is illustrated in Figure 7.4 (b).

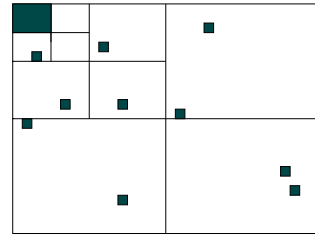
Policy 3: Mixture: low-pass plus maximum coefficients. A compromise would be to always synthesize the low-pass filtered part of a video which is put in the base layer l_0 . If the bandwidth has been selected above the minimum required to transmit l_0 , the remaining bandwidth is used as in policy 2. That is, the remaining bandwidth is used by successively defining thresholds with decreasing value, and by storing the coefficients of the wavelet-transformed domain whose absolute value is between two thresholds in the corresponding enhancement layer. This method is illustrated in Figure 7.4 (c).



(a) Policy 1. Blockwise: Inverse order of the decomposition.



(b) Policy 2. Maximum coefficients, no matter where they are located.



(c) Policy 3. Mixture: low-pass filtered part plus maximum coefficients.

Figure 7.4: Layering policies of a wavelet-transformed image with decomposition depth 3. The compression rate is set to 4.6875% (i.e., 3 blocks of the coarsest scale).

The three layering policies differ strongly in granularity. The blockwise policy is the coarsest one. One block at decomposition level 1 contains $1/4 = 25\%$ of information, a block at decomposition level 2 contains $1/16 = 6.25\%$, and a block at level 3 contains $1/64 = 1.5625\%$ of the information. Consequently, the granularity of the blockwise layering policy is restricted to the information levels 75%, 50%, 25%, 18.75%, 12.5%, 6.25%, 4.6875%, 3.125%, 1.5626%, etc. (see Table 7.4).

While policy 2 is almost infinitesimally scalable, the mixed policy requires the percentage of information to be at least as high as the size of the low-pass filtered part of the signal. When the percentage

is exactly as high as the low-pass filtered part, the mixed policy is identical to the blockwise policy¹.

The visual quality of policies 2 and 3 depends highly on the decomposition depth of the image. This results from the fact that coefficients in a wavelet space where no decomposition has been executed (or only a very rough one) still contain too much locality information. A low information percentage for synthesis might then result in many image pixels obtaining no information at all and thus staying gray. Figure 7.5 demonstrates this fact. Consequently, we claim that the iteration process shall be carried out as often as the parameter setting allows.



(a) Decomposition depth = 1.



(b) Decomposition depth = 2.

Figure 7.5: Frame 21 of the test sequence *Traffic*, decoded with the layering policy 2 at 6.25% of the information. (a) was synthesized after *one* decomposition step, (b) was synthesized with the *identical* amount of information, but after *two* decomposition steps. The original image is shown in Figure 7.7 (a).

7.5.2 Test Setup

We implemented all three layering policies in C++ at a Linux machine. Our test sequences contained 225 color frames and had the spatial resolution of 352×288 pixels (i.e., CIF format). The evaluation was carried out based on the orthogonal Daubechies wavelet filter banks from the filter length of 2 taps (i.e., Haar filter) to the filter length of 40 taps (i.e., Daub-20). Based on the results of our quality evaluation of different boundary policies for wavelet-encoded still images (see Section 6.3.2), we implemented circular convolution as the boundary policy. With circular convolution, however, the number of iterations possible on the approximation depends on the length of the filter banks. Only three iterations were possible for the longer Daubechies filters on our frames in CIF format. In order to get comparable results for our evaluation, we have stopped the number of decomposition levels for *all* filter banks at level three.

¹Example: as 1.5625% is just the size of the low-pass filtered part of the image at three decomposition levels — and no additional information is allowed — the results of policy 1 are identical to those of policy 3 according to the construction scheme (Table 7.4, last column).

7.5.3 Results

As explained above, our evaluation results on the performance of the ‘clever’ video metrics suggested we no longer use the ITS metric, and the remaining DIST metric yielded results comparable to those of the much simpler PSNR. Hence, the evaluation in this section is based only on the PSNR.

7.5.3.1 Visual Quality

The evaluation of the three layering policies was carried out with comparability in mind, i.e., at the percentages of synthesized coefficients that the blockwise layering policy meets (see Section 7.5.1). Again, we aimed to get an overview of the impact of different filter lengths, and therefore have opted to evaluate the filter banks: Haar, Daub-3, Daub-6, Daub-10, Daub-15, and Daub-20. Since the wavelet transform produces many detail coefficients close to zero for our input signals under consideration (which were not too noisy), the visual quality for the upper compression percentages (75%, 50%, and 25%) was excellent for all policies and all filter banks. Thus, Table 7.4 shows the evaluation of visual perception only for the more interesting lower information levels.

Quality of visual perception — PSNR [dB]									
Wavelet	Percentage of synthesized coefficients								
	18.75%			12.5%			6.25%		
	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3
Haar	47.185	69.257	69.210	43.892	63.085	63.008	41.004	54.628	54.385
Daub-3	47.260	68.347	68.311	44.468	62.024	61.956	40.988	53.535	53.280
Daub-6	48.393	67.111	67.073	45.225	60.887	60.835	42.079	52.89	52.723
Daub-10	47.958	65.215	65.183	44.923	59.087	59.018	41.802	51.052	50.863
Daub-15	48.664	64.312	64.273	45.339	58.388	58.313	41.717	50.796	50.593
Daub-20	48.295	62.992	62.960	45.153	57.173	57.101	41.656	49.816	49.627
average	47.959	66.205	66.168	44.833	60.107	60.039	41.541	52.12	51.912
Wavelet	4.6875%			3.125%			1.5625%		
	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3	pol. 1	pol. 2	pol. 3
Haar	40.570	51.505	51.088	39.047	47.341	46.435	35.210	40.882	35.210
Daub-3	40.609	50.596	50.190	39.214	46.685	45.899	37.235	40.757	37.235
Daub-6	41.640	49.969	49.599	40.077	46.275	45.602	37.041	41.253	37.041
Daub-10	41.372	48.428	48.133	39.701	45.272	44.743	36.734	40.441	36.734
Daub-15	41.291	48.176	47.850	39.644	44.951	44.370	36.817	40.136	36.817
Daub-20	41.237	47.371	47.096	39.610	44.371	43.880	36.882	40.038	36.882
average	41.120	49.341	48.993	39.549	45.816	45.155	36.653	40.585	36.651

Table 7.4: The PSNR of frame 21 of the test sequence *Traffic* for different decoding policies and different percentages of restored information. The best results of the PSNR within a given policy and percentage are highlighted. See also Figure 7.6 for a better visualization.

A closer look at the values of the PSNR in Table 7.4 shows that though the PSNR sometimes increases with increasing filter length, it decreases in other cases, notably for policies 2 and 3. This phenomenon appears only at a low information rate. An explanation might be that the synthesis of very little information in the transformed domain suffers from the location influence of long synthesis filters,

which ‘smear’ the incomplete signal information into neighboring locations. (cf. also the results in Section 6.3.5).

Figure 7.6 is based on the values of Table 7.4. The distinction between different wavelet filter banks has been removed and has been replaced by the average PSNR value of the six presented wavelet filters at the given percentage. It demonstrates that the visual perception of both policies 2 and 3 is very similar, and much better than the perception of the images synthesized blockwise.

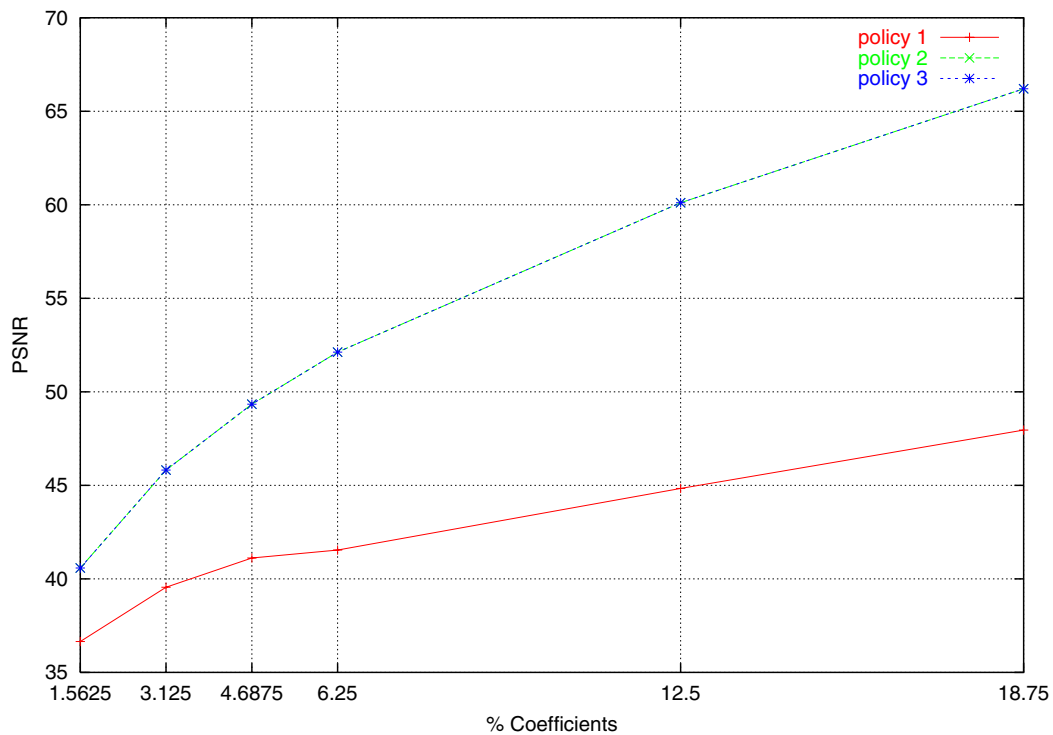


Figure 7.6: Average PSNR value of the Table 7.4 for different percentages of synthesized wavelet coefficients. While the perceived qualities of policies 2 and 3 are so close that both curves appear identical, policy 1 produces far lower quality.

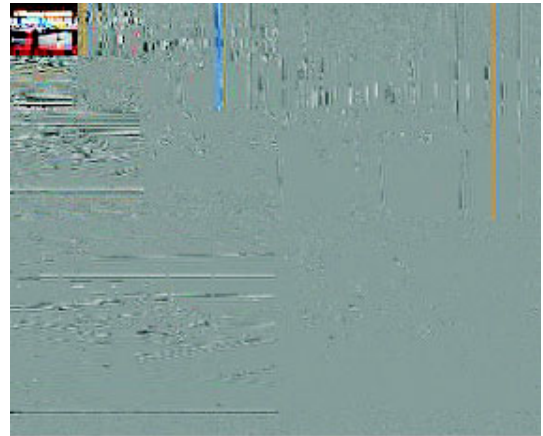
Figure 7.7 (a) shows frame 21 of our test sequence *Traffic*. This test sequence contains a lot of sharp edges (e.g., lantern, pile, house in background, advertisement ‘&’) while at the same time being composed of large uniform areas (e.g., house, cars, street, pavement). The frame has been decomposed to level 3 (b). While images (d) and (e) do not show large differences, (c) is clearly blurred. As both layering policies 2 and 3 allow the synthesis of detail information in low layers, the reconstructed image contains parts with high spatial resolution (i.e., sharp edges) — note especially the ‘&’ in the advertisement. In contrast, less important parts, such as the tree leaves, are resolved worse than in (c).

7.5.3.2 Bit Rate

Besides the visual quality, the bit rate produced by a layering policy is an important factor. The bit rate depends on the entropy encoding algorithm, though. DCT-based compression algorithms like JPEG



(a) Original frame.



(b) Wavelet-transformed.



(c) Policy 1: blockwise synthesis.



(d) Policy 2: maximum absolute coefficients.



(e) Policy 3: mixture of both.

Figure 7.7: Frame 21 of the test sequence *Traffic*. (a) shows the original frame. (b) visualizes the wavelet transform with a Daubechies-6 filter and decomposition depth 3. Images (c) to (e) show the syntheses of the transform with 6.25% of coefficient information.

and MPEG usually use run length and Huffman encoding in order to compress the DCT-transformed coefficients. Since our layering policies lead to a large number of zero-valued coefficients in the time-scale domain, run length encoding, applied to the quantized coefficients, is expected to result in good compression rates. Thus we suggest the following simple entropy encoding approach.

The coefficients in the wavelet-transformed domain contain different scale information. Within each scale, all coefficients are peers, i.e., they are at the same level of abstraction. In contrast, the different scales within the time-scale domain deserve different regard. Hence, the sampling of the coefficients in the time-scale domain is implemented in a line-by-line and subband-by-subband mode. Figure 7.8 illustrates this sampling approach. The thus sampled coefficients enter a run length encoding process, where a *run* z/n stands for an arbitrary number of $z + 1$ succeeding coefficients. The first z of those coefficients are zero-valued, while the $z + 1^{\text{st}}$ coefficient has a value of $n \neq 0$. With this approach, a sample sequence of the coefficients ‘300500001’ will be mapped to the data symbols $0/3$, $2/5$, and $4/1$. If the value zero is located at the end of a stream, the value $x/0$ is allotted.

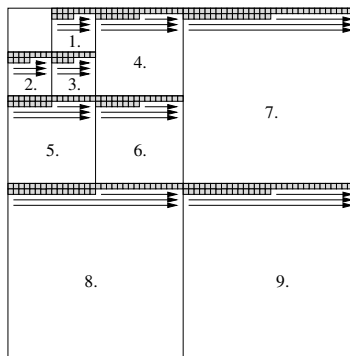


Figure 7.8: Linear sampling order of the coefficients in the time-scale domain.

Table 7.5 represents a heuristic for the de-facto bit rate of a layered coder: We use 6 bits to encode the run length z and 10 bits to encode the non-zero value n . Thus we need 16 bits to encode a single run. Table 7.5 shows the bit rate that resulted from each policy. It can be seen that the bit rates for the two best-quality layering policies, i.e., policies 2 and 3, are close together. Policy 3 wins the competition tightly. Concerning the choice of wavelet filters, the Haar wavelet produces considerably shorter runs and thus higher bit rates. The Daubechies-10 filter bank produces the longest runs and thus the lowest expected bit rate. Yet the Daubechies-3 filter bank with filter length 6 is sufficiently regular to result in a bit rate comparable to that for our longest test filter, Daubechies-20 with filter length 40.

7.5.4 Conclusion

In the above discussion, we have analyzed layered wavelet coding with regard to layering policy, scalability, visual quality, choice of orthogonal filter bank, and expected bit rate. We have detailed why we would not consider blockwise synthesis any further. Wavelet filter banks of 6 to 12 taps are advisable as shorter filters produce strong artifacts (see Table 7.4, Haar wavelet), and longer filters broaden the influence of erroneous synthesis at high compression rates (see Table 7.4, Daubechies-15 and Daubechies-20 wavelets). Finally, we have analyzed the expected bit rate for a single frame of a video sequence. Our tests state that the two layering policies 2 and 3 produce comparable bit rates,

Number of Runs (16 bit)						
Wavelet	Percentage of synthesized coefficients					
	18.75%		12.5%		6.25%	
	policy 2	policy 3	policy 2	policy 3	policy 2	policy 3
Haar	24443	24388	16885	16807	8511	8323
Daub-3	23588	23557	16095	16042	7945	7821
Daub-6	23178	23137	15747	15687	7821	7654
Daub-10	23006	22972	15521	15462	7619	7484
Daub-15	23258	23214	15736	15663	7742	7605
Daub-20	23359	23312	15804	15736	7887	7711

Table 7.5: Heuristics for the bit rate of a wavelet encoder for frame 21 of the test sequence *Traffic* with different wavelet filter banks. The lowest number of runs within a given policy and percentage is highlighted.

but policy 3 is expected to perform a little better. Taking into consideration that the scaling can be done at finer levels of granularity with policy 2 than with policy 3, we recommend to implement both layering policies and choose one depending on the context.

7.6 Hierarchical Video Coding with Motion-JPEG2000

In the previous section, we have discussed and evaluated the quality of a hierarchical video encoder with respect to the video quality, the choice of an orthogonal wavelet filter bank, the layering policy, and the expected bit rate. In this section, we present our implementation of a hierarchical motion-JPEG2000 video server and client. The parameters used for motion-JPEG2000 will be defined in part 3 of the JPEG2000 standard (see Section 6.4.1). At the current time, this part is still open for suggestions.

In contrast to the above discussion, the filter banks implemented were the reversible Daub-5/3 and the irreversible Daub-9/7 wavelet transforms (see Section 3.6). Due to the results of the layering policy in the previous section, we restricted the following considerations to the layering policy 3, i.e., the quantization of the coefficients in the wavelet-transformed domain is restricted to the high-pass filtered and band-pass filtered parts of the video, while the approximation is not quantized.

Furthermore, in extension of the calculation of the bit rate in the previous section, we have implemented a total of three different sampling schemes for the run length encoding after quantization in order to get empirical experience on whether the sampling order of the transformed coefficients influences the bit rate. As discussed before, the quantized and wavelet-transformed coefficients at each scale are stored in a fixed number of data symbols z/n which is defined through the number of quantized coefficients that are non-zero plus the optional border value $x/0$. The resulting *run length* of these data symbols, however, might depend on the sampling order of the coefficients: Varying with the structures of an image and the parameters used in the encoding process, the small, respectively, large coefficients, or the data symbols are concentrated in specific regions of the time-scale domain. A possible approach to reduce the actual bit rate of an encoded video could be to implement a sampling scheme that is especially suited to reduce the run length. In addition to the *linear* sampling of the

quantized time-scale coefficients for the run length encoding presented above, we have implemented two more sampling schemes into our video server: the *U-shaped*, and the *peano-shaped* sampling orders (see Figure 7.9).

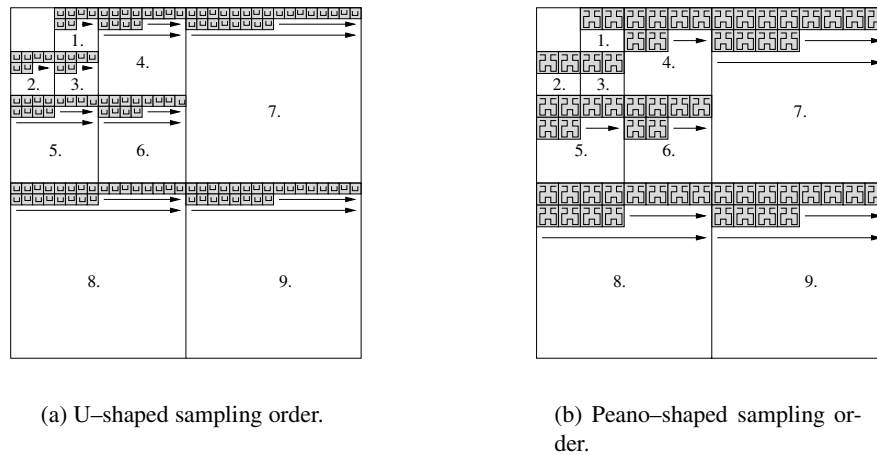


Figure 7.9: Sampling orders used by the encoder before run-length encoding.

The U-shaped sampling order is based on blocks of four coefficients, while the peano-shaped sampling uses units of 16 coefficients. No intrinsic reason suggests the two sampling orders in Figure 7.9, since the coefficients within a frequency band are peers. However, the idea is to sample larger blocks (in this context, the *linear* sampling could be interpreted as sampling blocks of only one coefficient). The above samplings form regular schemes. Together with the linear sampling order of the previous section, these three sampling schemes are proposed in [Str97] with the suggestion to use the peano-shaped sampling order.

7.6.1 Implementation

Our hierarchical motion-JPEG2000 video server and client were implemented as part of the master's thesis of Alexander Holzinger [Hol02] at our department. The programming language was Java 2 SDK, standard edition, version 1.3.1 with the Java Advanced Imaging (API) package. The program contains approximately 2300 lines of code.

The communication between server and client was realized by means of a DOS console. The server is started on a specific port, while the client has to specify both the IP address and the port of the server. A sample connection is the following:

Server on 134.155.48.11	Client on 134.155.30.40
java VideoServer 5656	java VideoClient 134.155.48.11 5656

Once the connection is established, the client opens a GUI which governs all further interaction of the user. Figure 7.10 shows the GUI of our motion-JPEG2000 client. The size of the grayscale frames

was set to 256×256 pixels. We used two home videos for our tests: The sequence *Mannheim* shows people walking around on the campus of the University of Mannheim; it contains 600 frames (i.e., 24 seconds). The sequence *Crossing* shows a street scene; it contains 650 frames (i.e., 26 seconds). The progress in the number of frames received and displayed by the client is indicated by a scrollbar.

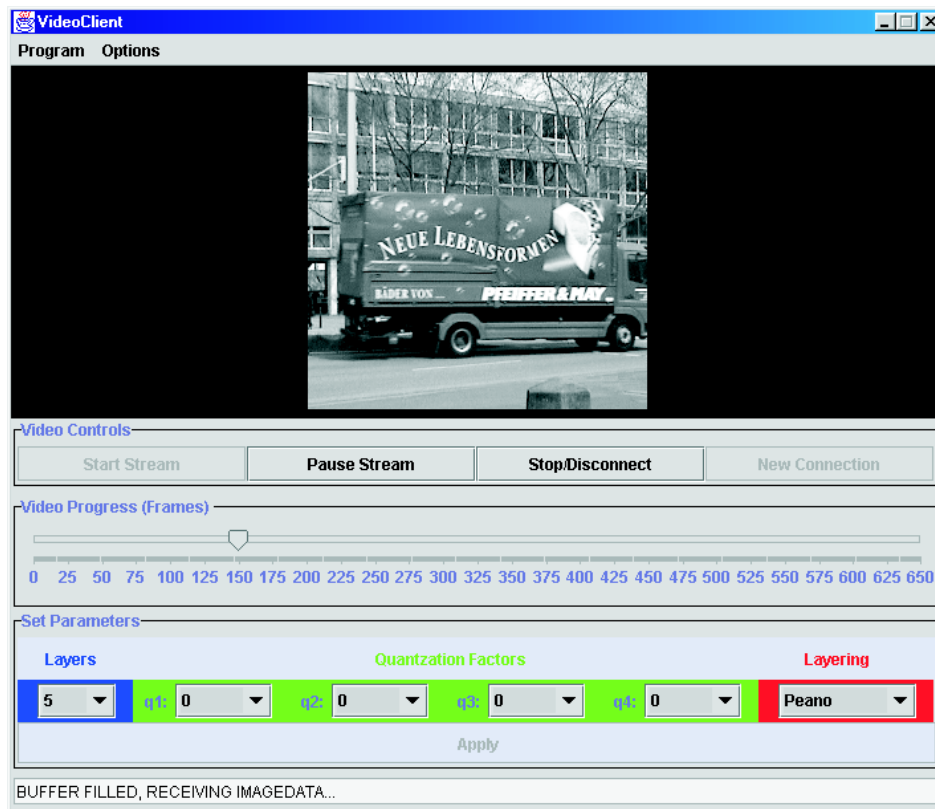


Figure 7.10: GUI of our motion-JPEG2000 video client.

The motion-JPEG2000 video server decomposes the video into five layers, where the base layer l_0 has the spatial size of 16×16 pixels. We have restricted our implementation to this decomposition depth for the sake of implementing the peano-shaped sampling order mentioned before. The number of layers received by the client and the corresponding quantization level q_1, \dots, q_4 for each of the enhancement layers l_1, \dots, l_4 has to be set manually on the client's GUI.

7.6.2 Experimental Setup

We were interested in the performance of the encoder and the decoder, as well as in the quality of the scaled grayscale video sequences. For this purpose, our video server and client were installed on different machines, connected with a 64 kbit/s ISDN line or a 100 Mbit/s LAN, respectively. We varied the three parameters

- sampling scheme of the time-scale coefficients,

- number of layers n_0 received by the client ($0 \leq n_0 \leq 4$), and
- quantization factors q_1, \dots, q_{n_0} applied to the enhancement layers l_1, \dots, l_{n_0} (if $1 \leq n_0$),

and measured the following four variables:

- number of frames actually received by the client,
- data received by the client (in bytes),
- duration of the transmission (in seconds), and
- objective visual quality (in dB by means of the PSNR).

The average number of bytes per frame as well as the data rate in Mbit/s could thus be easily derived from these variables.

7.6.3 Results

The following observations were obtained with a server installed on a pentium 3 with 666 Mhz and a client installed on a pentium 4 with 1.5 Ghz.

The setting of one of the sampling schemes: linear, U-shaped, or peano-shaped influenced our results on the above four variables only imperceptibly. In contrast to the suggestion in [Str97], our suggestion thus is that the selected sampling order has only a minor influence on the bit rate of a video stream. Hence, our further investigations were carried out with the linear sampling of the time-scale coefficients.

Table 7.6 details our results for an ISDN connection with one line (i.e., 64 kbit/s) for the video sequence *Crossing* with 650 frames. Obviously, the PSNR increases with an increasing number of transmitted layers. Furthermore, the PSNR reflects that the quality of the received video sequence depends on the quantization factors: With an increasing quantization factor, the PSNR generally decreases for a given number of layers. However, the visual quality of a transmission over an ISDN line generally is poor.

Two observations in Table 7.6 require an interpretation: the fact that frames got lost during the TCP/IP connection, and the very long duration of the transmission.

The loss of frames occurred predominantly when the quantization factor for the encoder was set high. The buffer of the client was then incapable of coping with some late-coming frames. The long transmission time is due primarily to the poor network connection. As soon as the amount of data is reduced drastically due to a different setting of the quantization thresholds or a different number of transmitted enhancement layers, the duration of the transmission shrinks as well. Table 7.6 shows good examples for this for the layers l_0 to l_2 .

Obviously, much better performance is expected of a better bandwidth connection between server and client. Table 7.7 shows our empirical results on a 10 Mbit/s LAN connection. Within a LAN, the loss of frames that can be observed in Table 7.7 for the transmission of all five layers can only be explained

Layers transmitted l_0, \dots, l_4	Quantization factors q_1, \dots, q_4	Frames received [number]	Data received [bytes]	Duration [sec]	Average [bytes/frame]	Average [kbit/s]	PSNR [dB]
l_0	—	650	335400	30	516	11	12.75
$l_0 + l_1$	$q_1 = 0$	650	1337700	133	2058	10	13.40
$l_0 + l_1$	$q_1 = 10$	650	1314920	120	2023	11	13.40
$l_0 + l_1$	$q_1 = 30$	650	964656	82	1484	11	13.39
$l_0 + l_1$	$q_1 = 50$	650	698536	61	1075	11	13.50
$l_0 + l_1$	$q_1 = 70$	649	560968	50	864	11	13.60
$l_0 + l_1$	$q_1 = 100$	649	426500	40	657	10	13.85
$l_0 + \dots + l_2$	$q_1 = 0, q_2 = 0$	650	5335200	500	8208	10	14.57
$l_0 + \dots + l_2$	$q_1 = 0, q_2 = 100$	650	1655244	162	2547	10	14.82
$l_0 + \dots + l_2$	$q_1 = 50, q_2 = 100$	650	1016080	89	1563	11	14.64
$l_0 + \dots + l_2$	$q_1 = 70, q_2 = 100$	650	879438	77	1353	11	14.50
$l_0 + \dots + l_2$	$q_1 = 100, q_2 = 0$	650	4424660	435	6807	10	14.29
$l_0 + \dots + l_2$	$q_1 = 100, q_2 = 50$	650	1706082	143	2625	12	14.26
$l_0 + \dots + l_2$	$q_1 = 100, q_2 = 70$	650	1099782	94	1692	11	14.36
$l_0 + \dots + l_2$	$q_1 = 100, q_2 = 100$	650	744704	67	1146	11	14.40
$l_0 + \dots + l_3$	$q_1, \dots, q_3 = 100$	646	2205626	188	3414	11	15.58
$l_0 + \dots + l_4$	$q_1, \dots, q_4 = 100$	639	4883278	455	7642	10	16.96

Table 7.6: Results of the performance evaluation for a 64 kbit/s ISDN line.

by a suboptimal server or client performance. As we have said above, our server was installed on a pentium 3 with 666 Mhz. This might be one of the reasons for the loss of video frames.

Regarding the time gap experienced by the client until the reception of the complete data stream, we see that the encoding of the video sequence into a base layer plus a single enhancement layer is a layering strategy that allows to receive the sequence in real-time (i.e., in 26 seconds), while the encoding into several enhancement layers requires a longer reception time. Again, the explanation is that neither our server and client nor the computers used for the evaluation were optimized. However, a maximum reception time of 69 seconds (with all five layers and no or only minimum quantization) indicates that our hierarchical video codec could be optimized for real-time applications.

7.6.4 Conclusion

Our quality evaluation of the performance of a motion-JPEG2000 hierarchical video codec states that current problems are due to not yet optimized test conditions and/or software. However, the principal ideas elaborated in this chapter on hierarchical video coding proved that the wavelet transform can be successfully exploited for this purpose. We have elaborated a number of recommendations on the parameter setting.

In contrast to many research efforts to measure the visual perception of a digital distorted video with intelligent metrics that reflect the human visual system, our extensive empirical evaluations have shown that — at least for our purpose — the peak signal-to-noise ratio performs sufficiently well.

Layers transmitted l_0, \dots, l_4	Quantization factors q_1, \dots, q_4	Frames received [number]	Data received [bytes]	Duration [sec]	Average [bytes/frame]	Average [Mbit/s]	PSNR [dB]
l_0	—	650	335400	26	516	0.10	12.75
$l_0 + l_1$	$q_1 = 0$	650	1337700	27	2058	0.40	13.40
$l_0 + l_1$	$q_1 = 10$	650	1314920	28	2023	0.38	13.40
$l_0 + l_1$	$q_1 = 50$	650	698530	27	1075	0.21	13.50
$l_0 + l_1$	$q_1 = 70$	650	561604	26	864	0.17	13.60
$l_0 + l_1$	$q_1 = 100$	650	427160	26	657	0.13	13.85
$l_0 + \dots + l_2$	0, 0	650	5335200	28	8208	1.52	14.57
$l_0 + \dots + l_2$	10, 10	650	5040060	28	7754	1.44	14.56
$l_0 + \dots + l_2$	30, 30	650	3284916	29	5054	0.91	14.46
$l_0 + \dots + l_2$	50, 50	650	1977458	27	3042	0.59	14.45
$l_0 + \dots + l_2$	100, 100	650	744704	28	1146	0.21	14.40
$l_0 + \dots + l_3$	0, 0, 0	650	21321300	30	32802	5.69	17.34
$l_0 + \dots + l_3$	10, 10, 10	650	18605572	32	28624	4.65	17.31
$l_0 + \dots + l_3$	10, 10, 50	650	10184668	31	15669	2.63	17.19
$l_0 + \dots + l_3$	10, 10, 100	650	6507646	29	10012	1.80	17.00
$l_0 + \dots + l_3$	10, 50, 100	650	4061428	29	6248	1.12	16.70
$l_0 + \dots + l_3$	10, 100, 100	650	3100050	29	4769	0.86	16.39
$l_0 + \dots + l_3$	50, 100, 100	650	2483666	29	3821	0.69	16.06
$l_0 + \dots + l_3$	100, 100, 100	650	2212290	29	3404	0.61	15.58
$l_0 + \dots + l_4$	0, 0, 0, 0	650	85254000	69	131160	9.88	identity
$l_0 + \dots + l_4$	0, 0, 0, 5	650	78451278	69	120694	9.10	44.35
$l_0 + \dots + l_4$	0, 0, 0, 10	646	66038444	67	102227	7.89	38.51
$l_0 + \dots + l_4$	0, 0, 0, 30	640	42576160	55	66525	6.19	29.57
$l_0 + \dots + l_4$	0, 0, 0, 50	642	32646390	51	50851	5.12	25.89
$l_0 + \dots + l_4$	0, 0, 0, 100	640	23753710	50	37115	3.80	21.88
$l_0 + \dots + l_4$	0, 10, 10, 10	633	63100680	62	99685	8.14	35.74
$l_0 + \dots + l_4$	0, 50, 50, 50	630	18771008	52	29795	2.89	22.25
$l_0 + \dots + l_4$	0, 100, 100, 100	620	5559217	41	8966	1.08	18.19
$l_0 + \dots + l_4$	10, 10, 10, 10	631	62930326	65	99731	7.75	34.99
$l_0 + \dots + l_4$	50, 50, 50, 50	620	17863392	52	28812	2.75	21.07
$l_0 + \dots + l_4$	100, 100, 100, 100	600	4539398	42	7566	0.86	16.96

Table 7.7: Results of the performance evaluation for a 10 Mbit/s LAN connection.

Furthermore, we have evaluated different strategies to subdivide a video stream into several quality layers. Our final evaluation of a motion-JPEG2000 video codec indicates that there is a high potential for wavelet-based hierarchical video encoding.

Until now, we have discussed the mathematical theory of wavelets, and several examples of applications to multimedia data streams. These applications were developed and evaluated in Mannheim as part of this dissertation. In our daily work with students at our university, we noticed that the theory of mathematical transformations and their application to signal analysis and compression are very difficult to understand. Only a small percentage of our students gained a deep understanding of the concepts, and of the influence of the different parameters of the algorithms. This has motivated us to investigate the potential of multimedia-based learning for this difficult material. In the context of our project VIROR (Virtuelle Hochschule Oberrhein) [VIR01], we spent a considerable amount of time and effort on the development and empirical evaluation of interactive learning tools for signal processing algorithms, and in particular on Java applets to be used in the classroom and for individual learning. This work will be reported in the next part of this dissertation.

Part III

Interactive Learning Tools for Signal Processing Algorithms

Chapter 8

Didactic Concept

This luke warmness arises [...] partly from the incredulity of mankind, who do not truly believe in anything new until they have had actual experience of it.

– **Niccolo Machiavelli**

8.1 Introduction

We have already mentioned that our engagement at the Department Praktische Informatik IV at the University of Mannheim was combined with the project on distance education: VIROR [VIR01]. The VIROR project aims to establish a prototype of a semi-virtual university and to acquire technical, instructional, and organizational experience with distance education.

This final part of this dissertation shifts the focus away from the technical background and its applications and instead discusses the teaching/learning aspect of source coding algorithms. In this context, issues such as

- *how to invoke the students' interest in source coding techniques and*
- *how to facilitate the hurdle to understanding complex topics*

illustrate the motivation to address the didactic aspects of distance education. Let us consider an example:

Example 8.1 *In our lecture 'Multimedia Technology', the JPEG coding standard is presented with its four steps: (1) image pre-processing, (2) discrete cosine transform, (3) run-length encoding, and (4) entropy encoding. In traditional teaching, our students were presented with the formulae of the one-dimensional and the two-dimensional discrete cosine transforms. As the exam at the end of a semester approached, our students would memorize that these formulae indicate the transition from the 'time domain' into the 'frequency domain'. However, very few students understood (a) what it means to*

analyze the frequencies in a given signal and (b) why this is done. It is this deep understanding of the underlying concepts which is the most important though.

The increasing popularity of multimedia-based distance education or *teleteaching* is reflected in the large number of projects dealing with this subject. We will be able to quote only a small number of them: Within the scope of the *Interactive Remote Instruction* project at the Old Dominion University in Virginia, a teleteaching system has been under development since 1993 that supports synchronous and asynchronous teaching scenarios [MAWO⁺97]. The *FernUniversität Hagen* in Germany has developed the platform *WebAssign* for the automation of weekly assignments: Submission, allocation, correction, and distribution of the assignments are assured via the Internet [Hag]. Other teleteaching projects include: *Teleteaching Erlangen–Nürnberg* [BBL⁺00], *Life Long Learning (L³)* [L3], *Universitärer Lehrverbund Informatik* [ULI] (all in Germany), *Distance Education at the University of South Africa (UNISA)* (South Africa) [CS00], and *Classroom 2000* at the Georgia Institute of Technology [Abo99].

Until today, the overwhelming majority of related teleteaching projects have concentrated on the technical fields (e.g., electrical engineering or computer science), sounding out and continually extending the technical options (see [ISLG00] [SHE00] [HSKV01] [BFNS00]). The didactic–pedagogical evaluation of such projects, however, examines the impact of the new learning environment on students:

1. Is computer-based learning an appropriate way to teach students?
2. Which qualitative statements on traditional learning in a lecture room as opposed to computer-based learning hold in general?

In cooperation between the departments *Praktische Informatik IV* and *Erziehungswissenschaft II* at the University of Mannheim, we worked on the definition, realization, and objective evaluation of pedagogic tools for interactive asynchronous teaching and learning.

In this chapter, we describe the didactic concept behind our teaching tools. Chapter 9 presents some of the implemented simulations and demonstrations which are related to the subject of the presented work. In the Summer Semester 2001, we performed a thorough evaluation of the topic *traditional learning versus multimedia-supported computer-based learning*, where the one- and the two-dimensional discrete cosine transforms were evaluated by over 100 students in different learning scenarios. The reason to use the discrete cosine transform rather than the wavelet transform was pragmatic: All test subjects were new to the topic of source coding, and the wavelet transform was considered to be too complicated for explanation within the scope of the evaluation. The results of the evaluation are detailed in Chapter 10.

8.2 The Learning Cycle in Distance Education

Determinant success factors of a lecture in the teleteaching scenario are the modularity of the lecture and the didactic concept behind the modules. For traditional learning scenarios in a classroom, lecturers often employ a series of still images to visualize a time-dependent topic. Their presentation then

resembles a flip-book, whereby the more complex a topic is, the more frames of still images it will involve, causing students to lose sight of the general idea.

Pedagogic evaluations have proven that a learner's capacity to imagine decreases with an increasing level of abstraction [HDHLR99] [HER⁺00] [HBH00] [Ker98]. Thus, a topic can be imagined and reproduced by a student only as long as its complexity does not exceed a certain level. Highly abstract themes, though, are not likely to be totally understood without any means of visualization [Ker98]. The better this visualization is, the greater the learning success.

In their unpublished project report of the *Learning through Telematics* project at the Open University, Mayes et. al. have introduced the *learning cycle* of a student in a distance education scenario [MCTM94]. This learning cycle is a prescriptive learning theory, subdivided into the three components *conceptualization*, *construction*, and *dialog* which mutually influence each other. The acquisition of one component is necessary to advance into the next one; the higher the level, the more profound is the acquired knowledge [GEE98] [HFH01] [MCTM94]. Figure 8.1 demonstrates the idea.

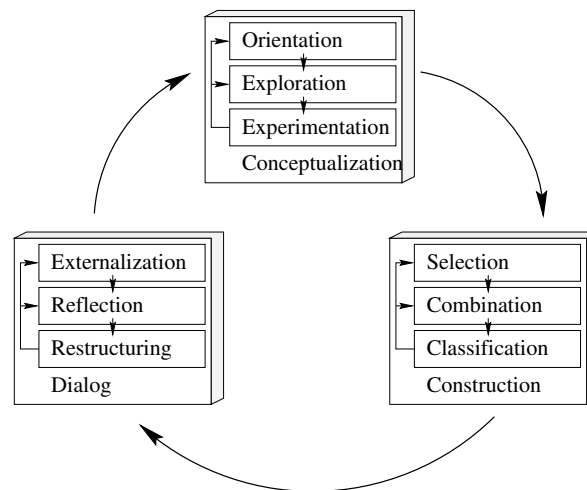


Figure 8.1: Learning cycle.

8.2.1 Conceptualization

This is the phase of *knowledge acquisition*. The aim is that knowledge shall be remembered not only for a short time, but in the long run. Thus, knowledge acquisition is combined with the orientation in, exploration of, and experimentation with the new topic. The learners come into contact with the concept of the underlying teaching material.

Requirements for the instructional media. Instructional media in this phase shall adequately introduce the field of reference, link to prior knowledge of the learner (better: test it), and clearly structure and present the topic. It is important to stick to a few central ideas without digressing into details.

8.2.2 Construction

This is the phase of *acquisition of problem-solving competence*. The learner shall make use of his/her newly acquired knowledge. This leads to a more profound understanding of the topic since the learner has to select, combine, and classify his/her knowledge according to its relevance to the actual problem.

Requirements for the instructional media. Instructional media in this phase have to allow the learner to intensively use the medium in a stand-alone fashion and at his/her own pace. The medium thus has to provide in-depth knowledge in response to the singular and unique questions of the learner; it must *not* be rigid or linear. The didactic challenge in this phase is to provide well-designed tasks and moderately difficult problems.

8.2.3 Dialog

The third phase is the *acquisition of meta knowledge*, i.e., the competence to decide issues such as *which concepts and procedures are adequate in which circumstances*. This is highly interconnected with the development of the learner's creative potential. Reflection and restructuring are important skills prerequisite to finally achieving the externalization of knowledge. With this last phase of the learning process, the model abandons the individual conception, but it takes into consideration the learning context and the social environment: significances are communicated and stabilized.

Requirements for the instructional media. Dialog is the ideal setting for human-computer interaction. Sample exercises of medium and high difficulty as well as a script with assigned tasks guide the reflection of the learner. These moments of externalization are of utmost importance in the software design since a topic can be labeled as 'fully understood' only if the learner is able to communicate it, thus putting it into a different context, and modifying it according to the situation. At the same time, the notion of 'dialog' abandons the context of human-computer dialog at this point, and enters into a general form of dialog where the achievement, however, is a challenge for computer-based tools.

Chapter 9

Java Applets Illustrating Mathematical Transformations

We shall not cease from exploration. And the end of all our exploring will be to arrive where we started, and see the place for the first time.
– T. S. Elliot

9.1 Introduction

This chapter presents some of the Java applets that were developed for the interactive teaching and learning of source coding algorithms. Java-based interactive demonstrations and simulations are a state-of-the-art technology which is helpful to overcome the didactic problems of learning mentioned before. They are a convenient tool as they (1) are accessible on the Internet via a standard Web browser and they require no special installation on the local platform, and (2) are platform-independent, i.e., a Java virtual machine allows the identical compiled `class` files to run under various operating systems.

The applets presented in this chapter realize the didactic postulates discussed in Section 8.2. For the implementations, we used Java 1.3 and the Swing GUI classes. An extensive help menu provides the beginner with close guidance, while the advanced student gains more and more freedom to explore the topic at his/her own pace and according to personal preferences. Furthermore, the following topics were realized:

- The GUI is written in English for the purpose of international usage.
- The structure of the applets is organized from left to right. That means that the order in which an applet has to be operated (e.g., display of original signal, parameter setting, display of modified signal) is structured in the same direction in which Westerners are used to reading.
- The corresponding contents of an applet are not only tied up with a graphical ‘frame’, but we make extensive use of background colors for semantic linkage.

The Java applets are used for *synchronous* teaching within a lecture or seminar as well as for *asynchronous* teaching. One of the main goals of the latter is to deliver educational material to students who can process it selectively and at their own pace depending on their learning preferences. In our university context, such material usually accompanies a lecture. A complete list of our teaching applets is stored at [SK01]. The applets on the one-dimensional discrete cosine transform (DCT) (see Section 9.3) and the two-dimensional DCT (see Section 9.4) were used for the evaluation presented in Chapter 10.

9.2 Still Image Segmentation

Since the human visual perception is strongly oriented towards *edges* [Gol89], the detection of edges within an image is a preponderant task for all major image processing applications. Most segmentation algorithms are based on edge detection (see Section 6.2), but compression algorithms also seek to extract edges in order to keep artifacts due to lossy compression as small as possible: In the compression process, the visually important edges ought to be maintained with maximal quality. Conversely, *textured* regions, i.e., regions with a fast-changing pattern, are particularly robust towards the detection of artifacts [Gol89]. Therefore, they might be coded in lower quality.

9.2.1 Technical Basis

We briefly review the intermediate steps of a smoothing pre-processor, some edge and texture detection algorithms, as well as the importance of thresholding.

Smoothing Pre-processor. Most digital images contain noise. A random pixel with no correlation to its neighboring pixels can affect an image such that the ‘essential’ information is occluded. Therefore, the application of a smoothing pre-processor (e.g., the *Gaussian* or the *median* filters) is a common practice in image coding.

- The Gaussian filter smoothes neighboring pixels by a weighted average. The filter coefficients are deduced from the Pascal triangle. Thus the 3-tap filter is $\frac{1}{4}[1\ 2\ 1]$, and the 5-tap filter is $\frac{1}{16}[1\ 4\ 6\ 4\ 1]$.
- The median filter extracts the pixel value that stands at the median position when the values that are covered by the filter mask are sorted. Our implementation uses a square filter mask of 3×3 or 5×5 pixels, thus the median is uniquely defined.

Edge Detection. *Edges* delimit objects from a background or define the boundary between two occluding objects. They are determined by a sudden change in pixel value. Therefore, many edge detection algorithms are based on the derivative and related ideas. In our applet, we have implemented the following edge detection algorithms: first derivative, second derivative, Roberts, Prewitt, Sobel, Robinson, Kirsch, and Canny [Par96] [Wee98] [GW93]. All but the last one employ a threshold to determine the edges of an image after convolution with the filter masks. The Canny edge detector is

more sophisticated: It combines smoothing (with a Gaussian filter) and edge detection. Canny requires the standard deviation of the Gaussian filter for the smoothing process, which is done separately in the horizontal and vertical dimensions, yielding two intermediate images. Edge detection is realized via the first derivative of the Gaussian filter. Two further steps are the *non-maximum suppression* and *hysteresis thresholding*. The latter requires two parameters, a low threshold and a high threshold. Consequently, the Canny detector requires three parameters instead of a single threshold.

Texture Detection. *Textures* describe the repeated occurrence of a pattern. The texture detection algorithms implemented in our applet are based on fractal dimension, gray level co-occurrence matrix (GLCM), and Law's Texture Energy. Details are given in [Par96].

Thresholding. Thresholding an image subdivides it into two regions and results in a binary mask: Every pixel of the image is compared to the selected threshold and is assigned a 'yes/no'-bit according to whether it lies above or below the threshold. Thresholding is used in edge detection and texture detection algorithms to determine whether a pixel belongs to the specified group or region. However, no exact definition of a 'good' threshold exists and its setting depends on the image as well as on the experience of the user (cf. the discussion on the determination of a threshold in Section 5.3.4).

9.2.2 Learning Goal

A student of edge and texture detection algorithms not only has to understand the algorithms themselves, and *why* a proposed algorithm is convenient; apart from these rather technical aspects, the student also has to include the determination of a convenient threshold in his considerations. The selection of a convenient threshold, however, depends on the image, on the selected algorithm, and on the purpose.

Thus, the goal of the applet on still image segmentation is to provide experience to the student so that he fully understands the concept of image segmentation and the use of pre-processing algorithms (see the learning cycle in Section 8.2). At the end, he should be able to answer questions such as:

Question	Ref. to Learning Cycle
· What is the idea behind a smoothing pre-processor?	Conceptualization
· What influence does the size of the filter mask have?	Conceptualization
· What is the influence of a threshold?	Construction
· How should the parameters be set?	Construction
· Why are edge detection and texture detection used?	Dialog
· What is the idea of derivative-based edge detectors?	Dialog

9.2.3 Implementation

Our image segmentation applet [Kra00] realizes the following structure:

- An image for the segmentation task can be selected from a pool of grayscale images.
- The Gaussian and the median smoothing pre-processors might be applied as often as suitable.
- Algorithms for both the edge detection and the texture detection might be selected.
- The thresholds for both algorithms might be adjusted.
- At the right hand side of the applet, the results of both the edge as well as the texture detections are displayed — based on the selected algorithm and threshold.

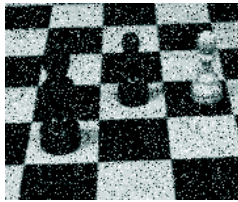
Thus, our image segmentation applet [Kra00] enables the user to experiment with all the different aspects of image segmentation algorithms and their parameters (see Figure 9.1). Three different background colors subdivide the three aspects of interest as follows: edges, texture, and background, i.e., the smoothed image minus edges minus texture. This simple color coding makes it intuitively clear which parameters influence what.



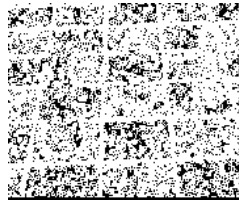
Figure 9.1: Graphical user interface of the segmentation applet. Here, the original image has been smoothed by a 3×3 median filter mask, the Canny edge detector has been applied, as has the GLCM mean texture detector with the threshold set to $\lambda = 90$.

With the help of the segmentation applet, our students were able to categorize visual phenomena and to make parameter setting recommendations. For example, Figure 9.2 (a)–(d) demonstrates the use of a smoothing pre-processor on noisy images before application of an edge detector. Figure 9.2 (e)–(h) demonstrates the outcome of different edge detection algorithms. Our experience shows that students

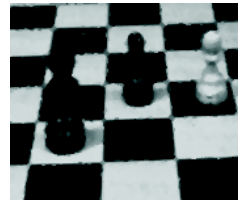
value this applet very highly since image segmentation is a very complex topic, not easy to understand from textbooks.



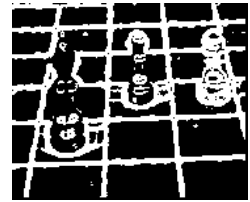
(a) Original image with noise.



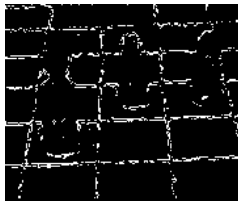
(b) Sobel edge detector, threshold $\lambda = 20$.



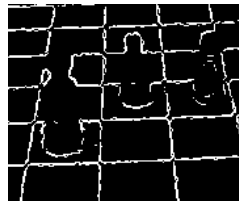
(c) Median filtering with mask 3×3 , applied twice.



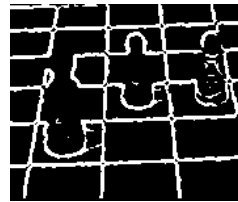
(d) Sobel edge detector on smoothed image, threshold $\lambda = 20$.



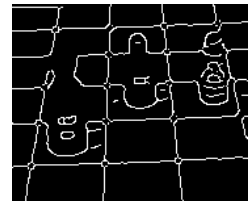
(e) Second derivative, threshold $\lambda = 70$.



(f) Roberts, threshold $\lambda = 70$.



(g) Prewitt, threshold $\lambda = 70$.



(h) Canny, $\sigma = 2.0$, low threshold $\lambda_l = 20$, high threshold $\lambda_h = 100$.

Figure 9.2: (a) – (d): Effect of smoothing on the edge detector. (e) – (h): Effects of different edge detectors.

9.3 One-dimensional Discrete Cosine Transform

The still image compression standard JPEG is based on the discrete cosine transform [PM93], which realizes the transition of a signal from the time domain into the frequency domain. This concept of transformation into the frequency domain is fundamental to the overwhelming majority of image compression algorithms. However, many students find the notion of frequency domain difficult upon their first encounter. This is one of the highly abstract subjects mentioned in Chapter 8, which requires visualization, demonstration, and experience for a better understanding on the part of the learner. Our DCT applet aims to stimulate the students' instinct for play as a way of dealing with the new concept. Its technical design and pedagogic purpose were presented in [SKW01]. The applet was also accepted for publication at the ACM Computer Science Teaching Center [ACM01], an online library for peer-reviewed electronic teaching and learning materials for computer science.

9.3.1 Technical Basis

The JPEG standard [PM93] defines that an image be subdivided into blocks of size 8×8 which are then transformed independently. In accordance with JPEG, we focus on samples of length 8, thus eight cosine frequencies form the basis in the frequency domain.

In traditional teaching, the one-dimensional discrete cosine transform is introduced by giving its formula

$$S(u) = \frac{c(u)}{2} \sum_{x=0}^7 s(x) \cos \left(\frac{(2x+1)u\pi}{16} \right), \quad (9.1)$$

where the normalization factor is $c(u) = \frac{1}{\sqrt{2}}$ for $u = 0$ and $c(u) = 1$ else. Here, $s(x)$ denotes the gray value of the signal at position x , and $S(u)$ denotes the amplitude of the frequency u . The inverse transform (IDCT), that takes values of the frequency domain and transfers them back into the time domain is given by

$$s(x) = \sum_{u=0}^7 \frac{c(u)}{2} S(u) \cos \left(\frac{(2x+1)u\pi}{16} \right),$$

where $c(u) = \frac{1}{\sqrt{2}}$ for $u = 0$ and $c(u) = 1$ else.

9.3.2 Learning Goal

The underlying concept of Equation (9.1) is that each periodic function can be approximated by a weighted sum of cosine functions of different frequencies. In traditional teaching, the DCT concept could be demonstrated by screenshots. Figure 9.3 shows two intermediate screenshots of the perfect approximation of the original curve (a) and (b), when the first two basis frequencies are put to their correct amplitude (c), and half of the frequencies are taken into account (d).

The learning goal of a student who shall comprehend the mathematical formula (9.1) is to understand these formulae and the rationale behind the application of the discrete cosine transform. At the end, he/she should be able to answer questions like

Question	Ref. to Learning Cycle
· Which are the predominant frequencies in a given signal?	Conceptualization
· Where does the notion of frequency enter into the formulae?	Conceptualization
· What is the significance of a basis frequency?	Construction
· Why are frequency transforms applied to digital signals?	Dialog

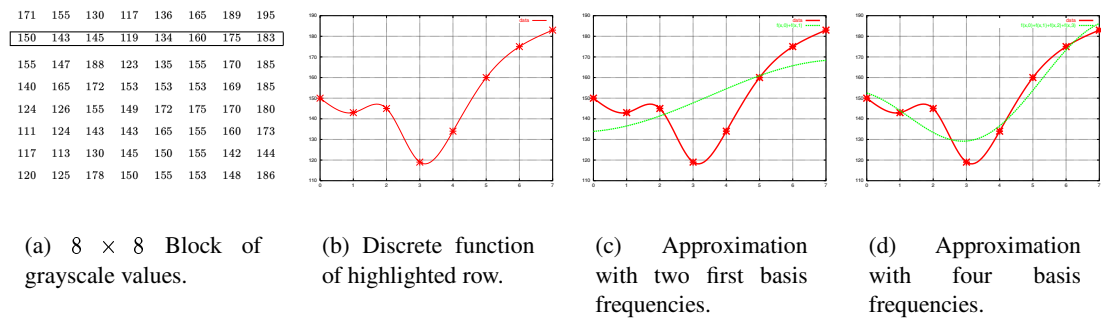


Figure 9.3: Figure (a) shows an 8×8 matrix with values of a grayscale image. The highlighted row is plotted against a coordinate system in (b) (for better visualization, the discrete points are linked via Bezier curves). (c) and (d): Subsequent approximation of the sample points by adding up the weighted frequencies.

9.3.3 Implementation

Figure 9.4 shows the GUI of our Java applet [Won00]. It is subdivided into two major parts:

- A *show panel* on the left hand side of the GUI shows a target signal that shall be approximated by setting the correct parameters in the frequency domain, and its approximation. Both the target and the approximation signal find two different visualizations: as a plot of the gray values in a graph (the target signal plotted in blue, and the approximation signal in red) as well as on a *color panel* (located under the graph), where the gray values of both signals are visualized.
- An *action panel* on the right hand side of the GUI accumulates the possible user interactions. It consists of an *input panel* where different pre-defined target values can be selected or changed, a *quantize panel* where the concept of quantization can be simulated, and — most important — the *scrollbar panel*. The approximation signal on the left results from the IDCT applied to the amplitudes of the cosine frequencies which are set with the scrollbars.

In order to motivate the learner, our applet guides the user by presenting him/her with the task of approximating a given target signal. Furthermore, the pre-defined target signals begin with very easy examples where only one single basis frequency is apparent, and get increasingly complex. Thus, the student is stimulated to play an *active* role by setting different values of the eight cosine frequencies with the scrollbars. An extensive help menu provides background information on the discrete cosine transform and on the usage of the applet. Finally, the perfect solution for the setting of the amplitudes for a specific approximation problem is provided as well, so that the student might have a look at the actual amplitudes of the basis frequencies for a given signal.

An aspect of the JPEG standard that we will not detail here is quantization. Amplitudes are quantized to reduce storage complexity. Our applet implements different quantization factors, where a chosen quantization is simulated by the direct effects on the inverse DCT: When quantization is applied, the amplitude coefficients are multiplied by the quantization factor before the IDCT is calculated.

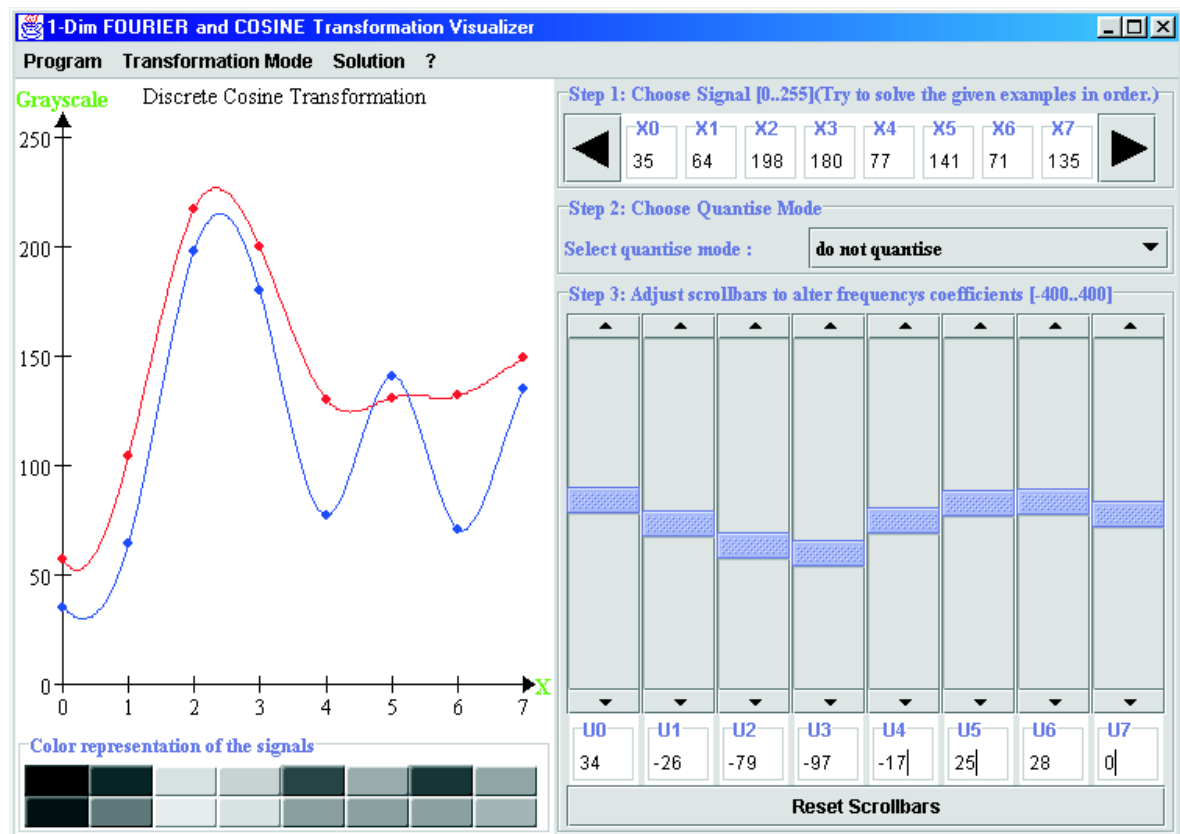


Figure 9.4: GUI of the DCT applet. In step 1, the user is asked to choose a (blue, here: lower) target signal from a given list or to build a new signal. In step 2, the challenge is to trim the scrollbars, i.e., the amplitudes of the eight basis frequencies, such that the IDCT (red, here: upper curve) behind these amplitudes approximates the given signal. A different presentation of both curves is given with the *color representation of the signals* below the plot. The upper gray row represents the target signal and the lower row the approximation.

9.4 Two-dimensional Discrete Cosine Transform

Our applet on the two-dimensional discrete cosine transform is an extension of the one-dimensional applet. The idea that the learner shall be incited to approximate a selected target signal by adjusting the amplitudes of the basis frequencies in the frequency domain has been adopted. However, this applet shows the gray values in numbers and as a plot, but not as a graph since a visualization of a graph with two variables (for the pixel position) and a gray value — hence, a three-dimensional graph — would be very complex without helping to illustrate the core idea.

The applet on the two-dimensional discrete cosine transform was also accepted for publication at the ACM Computer Science Teaching Center [ACM01], an online library for peer-reviewed electronic teaching and learning materials for computer science.

9.4.1 Technical Basis

Digital images are discrete two-dimensional signals, thus a two-dimensional transform is required to code them. Keeping the JPEG block size of 8×8 in mind, a single input element now consists of 64 pixel values which will be transformed into amplitudes for 64 cosine basis frequencies. The corresponding formula is given by

$$S(u, v) = \frac{c(u)c(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 s(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right), \quad (9.2)$$

where $c(u), c(v) = \frac{1}{\sqrt{2}}$ for $u, v = 0$ and 1 else. Here, $s(x, y)$ denotes the gray value of the pixel at position (x, y) and $S(u, v)$ denotes the amplitude of the two-dimensional frequency (u, v) . Analogous to the one-dimensional case, the inverse DCT is given by

$$s(x, y) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{c(u)c(v)}{4} S(u, v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right).$$

9.4.2 Learning Goal

Once the concept of a (one-dimensional) transform into the frequency domain is understood, the hardest part has been accomplished. However, most students have problems with two-dimensional imagination. It is also difficult to understand the notion of *frequency* in the context of a still image. Figure 9.5 shows some easy examples of images with a specific dominant frequency. While the examples (a) and (b) might be intuitively clear to most people, examples (c) and (d) are no longer obvious.

The purpose of the presented two-dimensional DCT applet is twofold:

- to furnish the user with the correct ‘feeling’ for a predominant frequency, so that the distinction between Figure 9.5 (c) and (d) becomes clear, and

- to help the user understand how the one-dimensional and the two-dimensional DCTs are related.

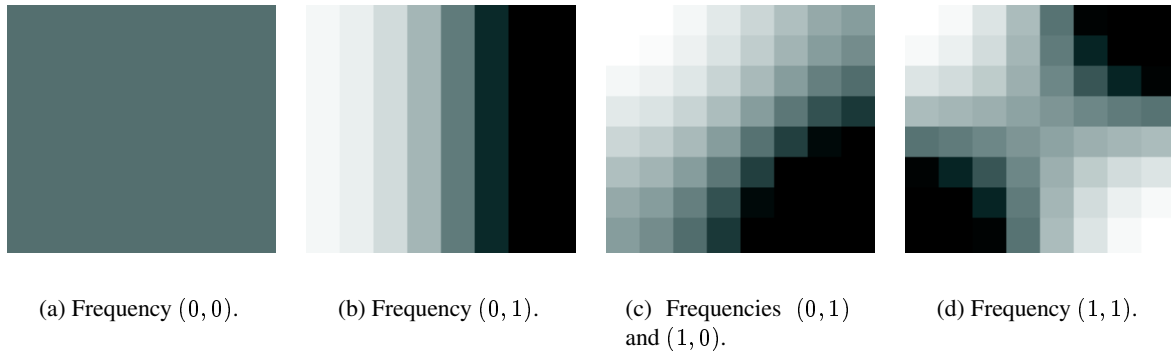


Figure 9.5: Examples of two-dimensional cosine basis frequencies.

9.4.3 Implementation

The GUI of this applet is shown in Figure 9.6. It is subdivided into two main parts: The left hand side shows the coefficients and the gray values of a target image, the approximation, and the difference signals, and the right hand side displays the adjustable amplitudes of the basis frequencies. Some pre-defined target signals are provided, arranged in the order of increasing difficulty. It is also possible to customize the target signal according to one's own preferences.

The amplitudes of the basis frequencies in the frequency domain can be set manually, or — similar to the one-dimensional applet — by means of a scrollbar. Since the two-dimensional DCT applet contains 64 basis frequencies, 64 scrollbars would have been a graphical overload on the GUI. This challenge has been met by equipping the GUI with just one single scrollbar, that nevertheless can be connected to each of the 64 basis frequencies by marking the frequency under consideration. The scrollbar itself is painted in yellow, as is the corresponding connected basis frequency, to make the connection obvious.

9.5 Wavelet Transform: Multiscale Analysis and Convolution

Theory and practice of the wavelet transform have been presented in Chapter 1. Especially the ideas of multiscale analysis (see Section 1.6) and convolution-based wavelet filtering (see the example of the Haar filter bank in Section 1.7 and the general case in Section 2.3) are complex ideas in need of illustration for the learner.

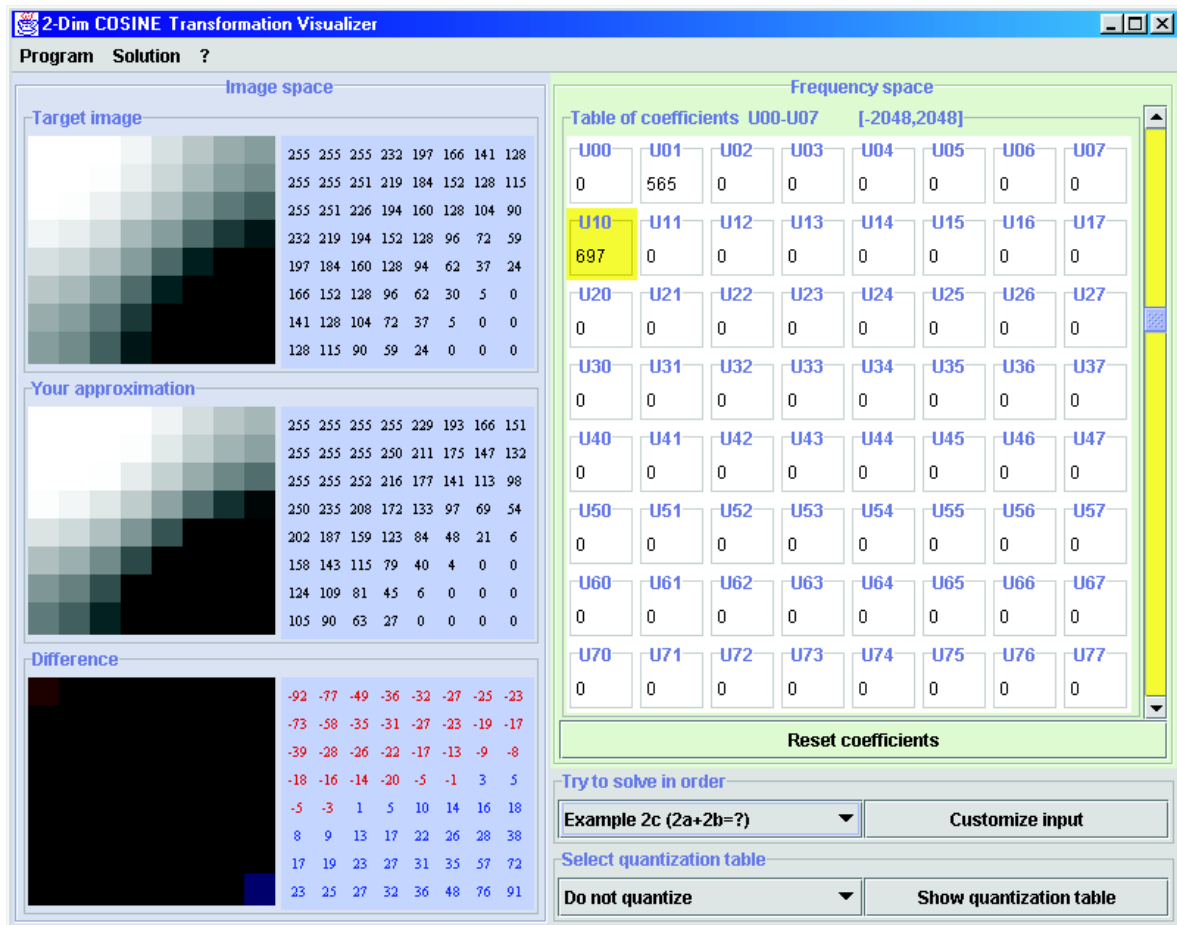


Figure 9.6: GUI of the 2D-DCT applet. The left hand side shows the selected target image (top), the approximation calculated by the inverse transform of the selected frequencies on the right hand side (middle) and a difference image (bottom). The user interface for the adjustment of the amplitudes is shown on the right hand side: The scrollbar on the far right is connected to a selected frequency, highlighted in yellow (here: gray). The lower right part of the applet contains additional functionalities.

9.5.1 Technical Basis

The technical basis of multiscale analysis and convolution-based wavelet filtering was presented in Part I. In Equation (1.4) we stated that the multiscale analysis requires two parameters: time and scale. In the approach of the dyadic wavelet transform, Equation (1.6) showed that the scaling parameter a can be set to multiples of the factor 2 without losing the property of perfect reconstruction. The translation parameter b , in contrast, governs the translation of the filter in the convolution process.

9.5.2 Learning Goal

Our experience shows that our students have great difficulty imagining a dilated and translated wavelet function. Understanding the convolution-based wavelet filtering as a transformation process is even harder for them. The problem is that the convolution of a signal with a filter bank is a dynamic process. Sketches like Figure 1.9 in Section 1.7 might help in the learning process, however, they are just a makeshift solution due to the lack of any better ones. A video, on the other hand, is not flexible enough to allow specific parameters to be set by the user.

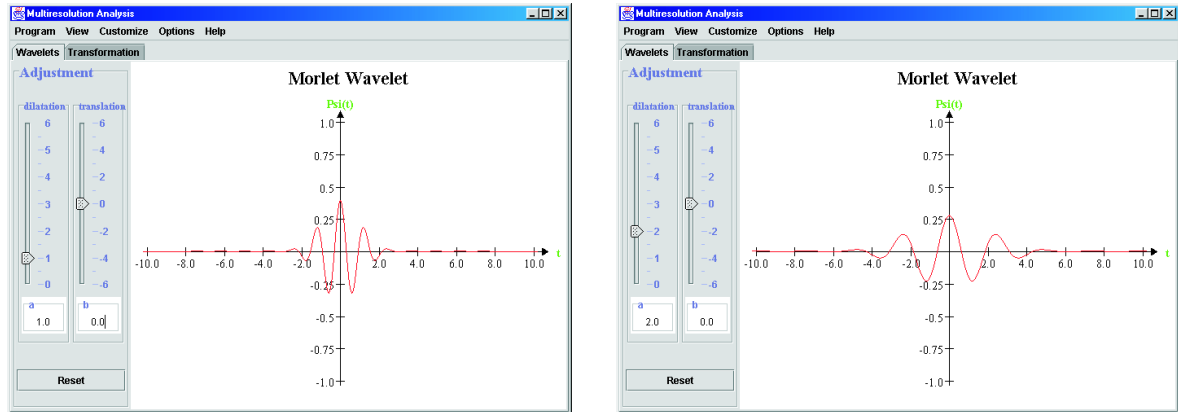
The learning goal of a student of the wavelet transform is to understand the significance of and the relation between the formulae and the implementation. The student should be able to answer questions like

Question	Ref. to Learning Cycle
· How does the dilation parameter influence the shape of a function?	Conceptualization
· How does the normalization parameter in the wavelet transform influence the shape of a function?	Conceptualization
· How does the translation parameter influence a function?	Conceptualization
· What is the relation between the dilation parameter and the notion of <i>scale</i> ?	Construction
· What is the relation between the translation parameter and the convolution process?	Construction
· How can the convolution process with the Daub-2 filter bank be generalized for arbitrary Daub- n filter banks?	Dialog

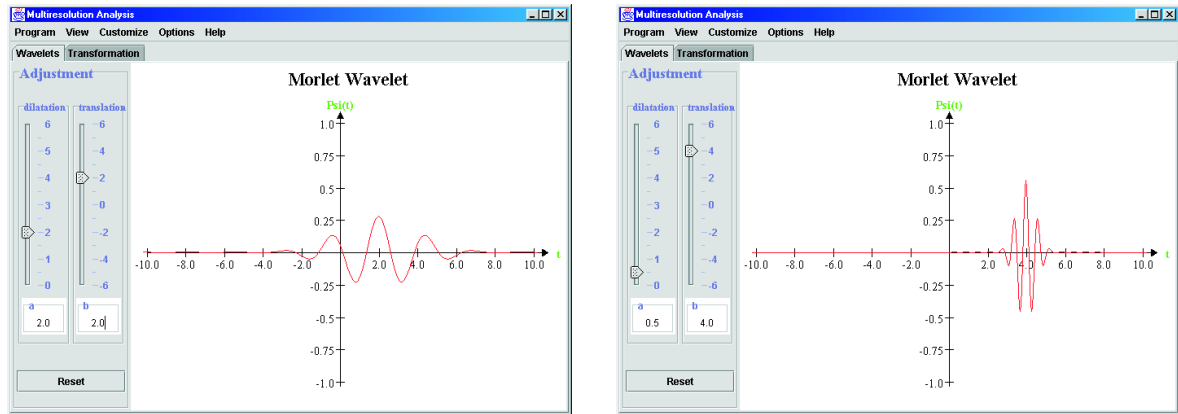
9.5.3 Implementation

Our applet developed for the demonstration of the multiscale analysis [Sch01a] holds a pool of wavelet functions: Haar wavelet, Mexican Hat wavelet, and the real part of the Morlet wavelet. These functions are displayed with the default settings of the dilation parameter a set to 1 and the translation parameter b set to 0. Each parameter can be varied, the dilation parameter between 0 and 6, while the translation parameter ranges from -6 to 6. Figures 9.7 (a) and (b) show screenshots of different parameter settings using the real part of the Morlet wavelet.

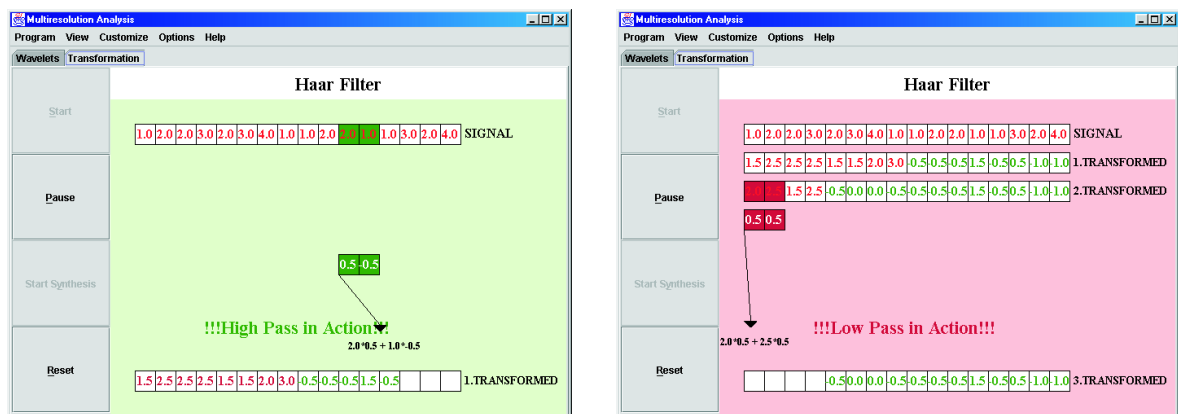
In the convolution process, a one-dimensional signal of even entries is wavelet-transformed with



(a) Multiscale analysis with different scale parameters (i.e., dilation).



(b) Multiscale analysis with different time parameters (i.e., translation).



(c) Convolution-based filtering of a one-dimensional signal with the Haar filter bank.

Figure 9.7: Applet on multiscale analysis with the real part of the Morlet wavelet ((a) and (b)) and on the convolution-based filtering process (c).

regard to either the Haar filter bank, or the Daubechies–2 filter bank [Sch01a]. Since the Haar filter bank has only two entries, no boundary problems arise. Yet the Daubechies–2 filter bank has to cope with the boundary problem. Once the boundary treatment for this filter bank of four coefficients is understood, the students shall be able to generalize the concept for orthogonal filter banks of arbitrary length. Hence, these two filter banks demonstrate an easy example (i.e., Haar) and a general example (i.e., Daub–2). Figure 9.7 (c) shows screenshots of the convolution process.

9.6 Wavelet Transform and JPEG2000 on Still Images

In this section we present an applet on the wavelet transform which enables the user to experiment on still images with all the different aspects of the discrete wavelet transform. The screenshots and images presented in Chapter 3 were also taken from this applet.

9.6.1 Technical Basis

Our applet on the wavelet transform demonstrates the effects of different settings for: image, filter bank, decomposition method, boundary policy, and quantization threshold, see Chapter 3. Its functionality is described in [SEK01].

9.6.2 Learning Goal

The learning goal for a student is to fully understand the concept of the wavelet transform, including the impact of parameter settings on the decoded image. At the end, he/she should be able to answer questions such as:

Question	Ref. to Learning Cycle
· What is the conceptual difference between standard and nonstandard decomposition?	Conceptualization
· What is the conceptual difference between the different boundary policies?	Conceptualization
· What is quantization? How is it used in the context of the wavelet transform?	Conceptualization
· What kinds of synthesis strategies exist?	Construction
· Why does the achievable iteration depth depend on the boundary policy?	Construction
· What visual impact do the different Daubechies– n wavelet filters have on the perception?	Construction
· What is the nature of a time–scale domain?	Dialog
· What influence do the parameter settings have on the decomposition process and image quality?	Dialog
· What are the strengths and weaknesses of the wavelet transform?	Dialog

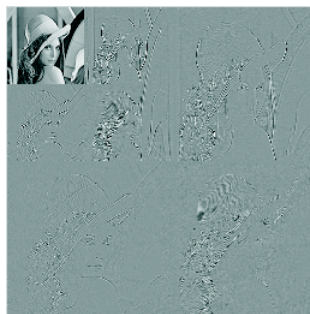
9.6.3 Implementation

Our wavelet transform applet [Ess01] has two different modes:

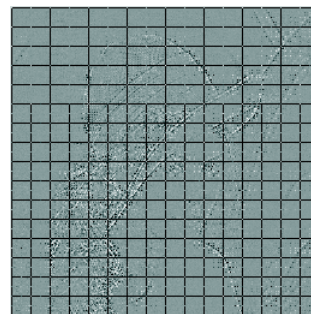
- convolution mode, and
- JPEG2000 mode.

In the convolution mode, the Haar wavelet filter bank, and the Daubechies filter banks of 4, 6, 8, 10, 20, 30, and 40 taps are supported. According to the selected wavelet filter bank and boundary policy, the number of iterations is carried out as often as possible (see Section 3.3.3).

When the JPEG2000 mode of the applet is selected, the border extension is set to mirror padding. The two standard lifting-based filter banks Daub-5/3 and Daub-9/7 are proposed (see Section 3.6), and the display mode of the coefficients in the time-scale domain can be set to either ‘separated’, i.e., the approximations and the details are separated in the time-scale domain (see Figure 9.8 (a)), or to ‘interleaved’, i.e., the coefficients are stored in the interleaved mode suggested by the standard (see Figure 9.8 (b)).



(a) Separated display mode: approximations and details are physically separated.

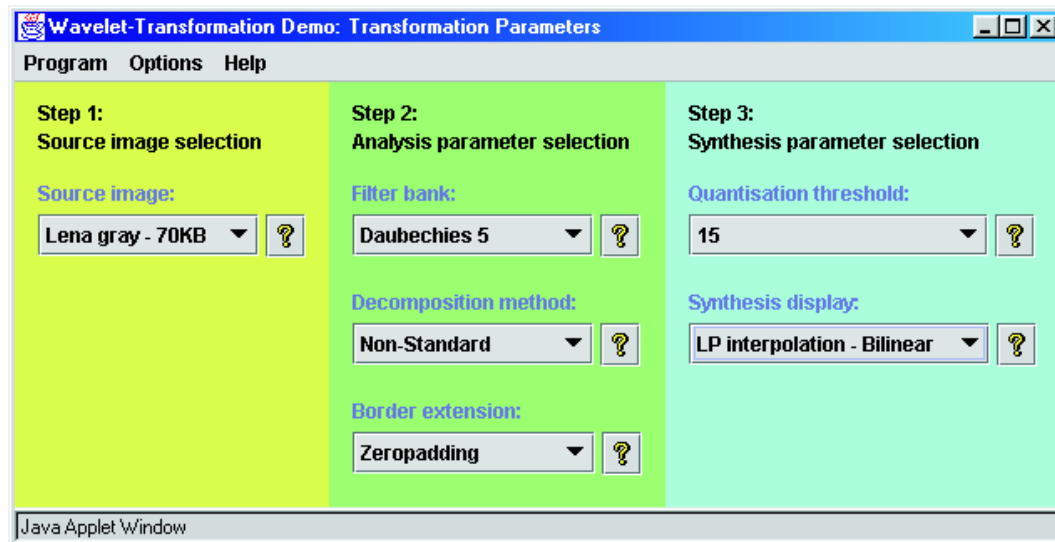


(b) Interleaved display mode: in JPEG2000, approximations and details are stored interleaved.

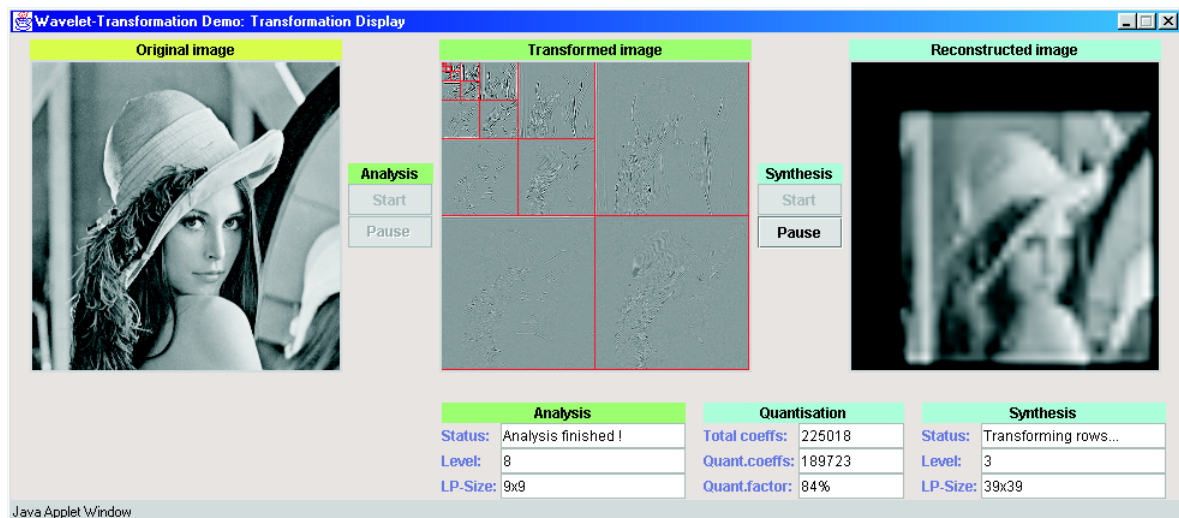
Figure 9.8: Different display modes for the time-scale coefficients.

In either mode, the GUI of our applet is divided into two parts, the *parameter definition* window (see Figure 9.9 (a)) and the *transform visualization* window (see Figure 9.9 (b)). The parameter definition window allows to select the image, the filter bank, the decomposition method and the boundary policy, respectively, to choose a separated/interleaved display mode for JPEG2000. A quantization threshold on the wavelet-transformed coefficients governs the quality of the decoded image. Different display modes for the decoded image exist to display the image even if the decoder has not yet received the complete information (see Section 3.5). The transform visualization window contains the selected image, the time-scale domain, and the decoded image.

Each window is structured from left to right. The three different background colors in the parameter definition window are also found in the transform visualization window. They indicate the subdivision



(a) Parameter definition.



(b) Transform visualization.

Figure 9.9: The two windows of the wavelet transform applet used on still images.

of parameters and visualization into the following fields: *source image*, *analysis*, and *synthesis*. This simple color scheme makes it intuitively clear that the parameters set in one window influence that specific part of the transformation in the other window.

9.6.4 Feedback

We put this applet on the Internet in late 2000 [SE01]. At the time of this writing, we have received 18 emails from all over the world with very positive feedback on our applet and the request for its source code. We cite an example (translated from German):

[...] since we just addressed the wavelet compression in the subject *media & media streams*, I have found your Web page on teaching applets. Our lecture unfortunately could not explain how the wavelet transform and the wavelet compression work, nor the underlying ideas (not to say that actually, only formulae were thrown at us, without any explication of the circumstances). Due to the applet and the documentation, the real facts have been concisely pointed out to me. The applet is really fantastic! It is very intuitive from the start, and the graphics are pleasing. What has helped me particularly? At the right moments, it voices the point of view of image processing. Hence, I suddenly realized how this works. Namely, I had thought for weeks about the meaning of a high-pass, respectively, a low-pass filter. Before, I had only heard of them in the context of audio analysis. I thus would be interested in the source code as well [...]

Chapter 10

Empirical Evaluation of *Interactive Media in Teaching*

Teaching should be such that what is offered is perceived as a valuable gift and not as a hard duty.

– Albert Einstein

10.1 Introduction

The Java applets presented in the previous section were used at the University of Mannheim for demonstration within a classroom and were provided as well to the students for asynchronous learning according to their own learning preferences and at their own pace. Motivated by the two questions

- *can a good computer-based training outperform a ‘traditional’ lecture held by a professor in a lecture hall?, and*
- *what are the didactic conditions that influence the learning success and failure of distance education tools?,*

we evaluated the effectiveness of the applets on the discrete cosine transform (see Sections 9.3 and 9.4) on 115 students of computer science. As mentioned before, our experience was that our students found it quite difficult to understand this topic. Thus, we were curious to what extent these multimedia applets, which allow a hands-on experience, would help our students.

A reference group of students attended a lecture. The majority of the test subjects, however, enjoyed our computer-based learning programs, where this latter group was further subdivided into various subgroups in different didactic surroundings. This empirical evaluation was carried out in cooperation with the department Erziehungswissenschaft II of the University of Mannheim in June 2001. The results were published in [SHF01].

10.2 Test Setup

Only students enrolled in computer science were selected to participate in this evaluation since this guaranteed that they would have the background to understand the purpose and use of the discrete cosine transform. Nonetheless, the students were just beginning their studies. Since coding standards enter the curriculum of Mannheim students of computer science only during their third or fourth year of study, all students had the same level of prior knowledge: none. This resulted in a homogeneous test group.

10.2.1 Learning Setting

A total time of 90 minutes was allotted for each learning setting; in each instance a central 60-minute block of learning time was preceded by a 15-minute preliminary test (see Appendix A.2.1) to record sociodemographic variables and information on covariates such as preliminary knowledge, and followed by a follow-up test (see Appendix A.2.2) to gather dependent variables.

10.2.1.1 Traditional Learning

The traditional lecture was held by Prof. W. Effelsberg with color transparencies on an overhead projector. Students generally like his lectures very much as they are clearly structured and he has a nice manner of presenting, always combined with a few jokes, but never losing sight of the general idea. In Section 10.3.2 we will see that on a scale from 0 (*trifft nicht zu*, i.e., does not apply) to 3 (*trifft zu*, i.e., does apply), the lecture of Prof. Effelsberg was rated at an average of 2.317 points, which is very good. Nevertheless, having just begun their studies, our test candidates in the lecture hall were unacquainted with him, so that they encountered the lecture on *Compression Techniques and the DCT* unbiased.

10.2.1.2 Computer-based Learning

For the test candidates in the computer-based learning scenario the central 60-minute learning block was divided into three 20-minute periods, each allotted to one of the three modules each candidate received:

1. introductory video (encoded as `real`),
2. applet on the one-dimensional DCT (see Section 9.3),
3. applet on the two-dimensional DCT (see Section 9.4).

The programs were installed on laptops equipped with a headset. The digital video showed Prof. Effelsberg giving an introduction to *Compression Techniques and the DCT*, this time only as an oral presentation without any visuals. In the video, Prof. Effelsberg welcomes the student, introduces the evaluation, and proceeds to address the compression topic. Half of the video is dedicated to

instructions on how to use the two applets on the discrete cosine transform. Figure 10.1 shows photos taken during the evaluation.



(a) Groups of candidates in their learning position.



(b) Details of the multimedia-based learning settings.

Figure 10.1: Photos of the evaluation of the computer-based learning setting.

The learning cycle (see Section 8.2) in our computer-based learning setting was implemented as follows:

- Conceptualization: introductory video.
- Construction: use of the programs with very strong guidance, i.e., extensive use of the help menu.
- Dialog: The guidance within the program becomes insignificant; the student works exclusively with examples.

This externalization of knowledge was especially supported by the *scripted* learning setting (see below), in which the students were incited to use the knowledge acquired to solve more general questions.

10.2.2 Hypotheses

The large number of 115 participants allowed us to test two important hypotheses on the effect of different *learning instructions*, i.e., information about the purpose of the programs, as well as on the effect of different *attribution*, i.e., information about the provenience of the programs, on the knowledge gained during the learning period.

- *Guided learning by means of a script*¹. If one wants to deal with a new problem, a part of one's attention is directed towards the topic itself, but another share of attention is used to hook up with the learning context. *Cognitive load* denotes the capacity which must be spent by the working memory in order to use a learning environment. The higher the cognitive load of a learning environment is, the less capacity is left for the treatment of its topic [Swe88] [Swe94]. A *script* is an example of an instruction to deepen the contextual dispute. It might thus lower the cognitive load of non-contextual elements of the program [BS89] [Ren97] and facilitate the learning process in complex learning environments. Since a new computer-based learning program generally requires a cognitive load, one expects to reject hypothesis $H_{1;0}$:

$H_{1;0}$: There is no difference in the objective knowledge the students gain from a traditional lecture or from computer-based learning.

$H_{1;1}$: There is a difference in the objective knowledge the students gain from a traditional lecture or from computer-based learning.

- *Pygmalion effect*. The notion of a *pygmalion-effect* entails a difference in the learning outcome as a result of different expectations of the teachers towards the students (and the reverse!), depending on the teacher's or student's anticipation of the learning, respectively, teaching quality [RBH74] [Jus86] [Hof97]. In analogy to the role of the teacher in a learning process, a positive, respectively, negative anticipation of a computer-based program's teaching quality is expected to yield different subjective ratings and different learning results. Due to the pygmalion-effect, it is expected that both the learning effect and subjective rating of a computer program will be higher if a student *assumes* the program to be of better quality. Hence, one expects to reject hypothesis $H_{2;0}$:

$H_{2;0}$: There is no difference in the objective knowledge the students gain from different settings of computer-based learning.

$H_{2;1}$: There is a difference in the objective knowledge the students gain from different settings of computer-based learning.

The independent variables *IV* in our evaluation, i.e., the factors of the test setting for the above hypotheses, were set as follows. In order to test the above hypotheses, the *computer-based learning* setting of the students was further subdivided into four groups: *exploration*, *script*, β -*version*, and *c't-article* (cf. the original documents in Appendix A.1). Together with the group in the traditional lecture-based learning scenario, this totaled five different test settings.

¹A *script* denotes a set of questions on the learning topic. It includes questions of both easy and moderate difficulty, thus allowing a *guided* study. The script used in our empirical evaluation is given in Appendix A.1.2 (in German).

IV_1 . Test of hypothesis $H_{1;0}$:

- **Lecture**: One group of students attended a traditional 60-minute lecture.
- **Exploration**: The students in this computer-based learning scenario were told to explore the learning environment without any exertion of influence in order to learn about the topic (see Appendix A.1.1). They were given no additional information about the provenience or purpose of the programs. Since this setting is the usual way of working with a computer-based training, this group serves as the *reference group* in our evaluation.
- **Script**: The students in this scenario were told to prepare the contents of what they will learn as if they should later present the topic to a fellow student. The students were provided with a script of sample questions as a guideline for their preparation (see Appendix A.1.2).

IV_2 . Test of hypothesis $H_{2;0}$:

- **Exploration**: The *reference group* in our evaluation, see above.
- **β -version**: The students were told that the computer-based learning programs had been developed as part of a *Studienarbeit*. A *Studienarbeit* is implementation work by a student which every student has to do during his/her studies, but which is not graded. Thus, the quality of such implementations is often moderate, i.e., β -version (see Appendix A.1.3).
- **c't-article²**: With the kind permission of the c't impressum, we distributed a (false) 'preprint' of the next issue of the c't magazine, in which our computer-based learning software was lauded as one of the best examples of computer-based training worldwide (see Appendix A.1.4).

The students were blind to their assignment to one of the five settings. Especially the split-up between the lecture and one of the settings of computer-based learning was carried out in a manner intransparent to them. They knew only the time at which they should arrive at a particular location. This method precluded selection of a scenario according to one's own preferences.

The dependent variables in the evaluation, i.e., the variables that were influenced by the test settings, included: follow-up knowledge test, subjective rating of the learning environment, mood before and after the test, self-esteem after success, and self-esteem after failure.

The covariates, i.e., the variables which are not influenced by the test setting, but which might influence the dependent variables, included: preliminary knowledge test and average grade in prior exams.

10.3 Results

The test results can be classified into two groups:

- **Descriptive Statistics**: a listing of data.

²The c't is a German magazine on computers and technology of a very high standard of quality.

- *Analysis of Variance*: Not all data are significant for the explanation of a fact. The analysis of variance compares the different cells and asks for *significant* variance, i.e., for a high probability to justifiably assume $H_{1;1}$ and $H_{2;1}$ and thus a low probability p to *wrongly* discard the $H_{1;0}$ and $H_{2;0}$ hypotheses.

In the following, we will detail our results on both analyses.

10.3.1 Descriptive Statistics

Table 10.1 shows the descriptive data of the probands. This allowed to take into consideration biasing covariates before interpreting the results. The students of computer science are predominantly male, which is reflected in the share of 87% male probands. They are all at the beginning of their studies, which is reflected by the age range of 18 to 25 with an average age of 21.22 years. The semester of study varies, but a standard deviation of 1.42 on the average of 2.51 semesters (i.e., less than 1.5 years) confirms that the overwhelming majority were at the beginning of their studies. The average amount of computer usage in hours per week sums up to 20 hours for private and non-private usage.

	N	min	max	mean \bar{x}	std. dev.
Age	115	18	25	21.22	1.18
Semester	115	1	12	2.51	1.42
# Years of non-private computer usage	115	1	12	4.75	2.56
# Years of private computer usage	115	0	16	8.30	3.48
# Hours per week of non-private comp. usage	115	1	40	7.27	6.01
# Hours per week of private computer usage	115	1	50	12.79	8.71
Preliminary knowledge test	115	1	7	4.64	1.43
Average grade on prior exams	110	1.1	5.0	2.92	0.99
Mood before versus after test	114	2.25	5.00	3.90	0.60
Mean rating	114	0.67	3.00	2.36	0.53
Follow-up knowledge test	113	1	9	6.05	1.81
Valid Entries (per list)	107				

Table 10.1: Descriptive statistics on the probands.

The test on pre-existing knowledge contained seven multiple-choice questions on the general context of the student's current lecture as well as on the fundamentals of image compression (see Appendix A.2.1), resulting in a possible score of 0 to 7 points. The follow-up knowledge test contained nine questions to measure what the students learned during the 60-minute learning period (see Appendix A.2.2). Originally, the follow-up test encompassed a tenth question as well, but the answer to this question led to misunderstandings, and we decided to withdraw it from further evaluation. The average grade on prior exams asked for the students' grades on exams they had taken in their first semester. In Germany, 1.0 (i.e., 'sehr gut') is the best possible grade and 5.0 (i.e., 'mangelhaft') means 'failed'. This covariate takes into consideration the general ease of learning of the specific test candidate.

	Setting	N	mean \bar{x}	std. dev.
# Years of total computer usage	<i>Lecture</i>	28	6.8929	3.2385
	<i>c't-article</i>	19	5.8684 [†]	2.5919
	<i>β-version</i>	21	6.0952	2.1072
	<i>Script</i>	22	7.0227 [†]	2.7320
	<i>Exploration</i>	17	6.7941	2.2225
	Total	107	6.5654	2.6618
# Hours per week of total comp. usage	<i>Lecture</i>	28	19.3929	10.0493
	<i>c't-article</i>	19	22.4211 [†]	12.7249
	<i>β-version</i>	21	17.1429 [†]	10.0364
	<i>Script</i>	22	21.7955	12.2753
	<i>Exploration</i>	17	18.4118	9.2740
	Total	107	19.8271	10.9027
Preliminary knowledge test	<i>Lecture</i>	28	5.0000 [†]	1.5870
	<i>c't-article</i>	19	4.2632	1.2842
	<i>β-version</i>	21	4.8571	1.6213
	<i>Script</i>	22	4.5455 [†]	1.2994
	<i>Exploration</i>	17	4.7647 [†]	1.2515
	Total	107	4.7103	1.4341
Average grade on prior exams	<i>Lecture</i>	28	2.8905	1.0004
	<i>c't-article</i>	19	2.8632	0.9815
	<i>β-version</i>	21	2.9413	1.2630
	<i>Script</i>	22	3.1295 [†]	0.9716
	<i>Exploration</i>	17	2.6147 [†]	0.6892
	Total	107	2.9009	1.0014
Follow-up knowledge test	<i>Lecture</i>	28	6.5536	1.4164
	<i>c't-article</i>	19	6.4474	1.5977
	<i>β-version</i>	21	4.9524 [†]	1.8433
	<i>Script</i>	22	6.9091 [†]	1.4931
	<i>Exploration</i>	17	5.7647	1.8718
	Total	107	6.1682	1.7442

Table 10.2: Descriptive statistics on the probands, detailed for the setting. The table entries are limited to the 107 valid entries. The cells marked with [†] are explained in the text of Section 10.3.1.

Table 10.2 details the five most important entries of Table 10.1 on the different learning settings. Here, it becomes obvious that the distribution of the students over the settings was not uniform; a fact that is considered in the analysis of variance (see below). The most important differences were:

- *‡ Years of total computer usage.* The students in the setting *c't-article* had used the computer for only 5.87 years, compared to 7.02 years by the students in the setting *script*. This means that the prior knowledge of computer handling was 11% less than average (6.57 years) and 16% below the *script* group. Thus, the initial configuration did not point out the fact that these two groups would perform especially well.
- *‡ Hours per week of total computer usage.* For hypothesis $H_{2,0}$, the two settings of *c't-article* and β -version were in direct competition. However, these two groups also make up the maximum (22.42, *c't-article*) and the minimum (17.14, β -version) of total computer usage per week.
- *Preliminary knowledge test.* The differences between the three groups *lecture* with 5.00, *script* with 4.54, and *exploration* with 4.76 points on the preliminary test are not only astonishing, but Table 10.3 reveals that these differences are also significant. Thus, the students attending the lecture had a *significantly* better starting position. Yet, the results in Section 10.3.2 show that the *scripted* computer-based learning outperformed the *lecture* setting.
- *Average grade on prior exams.* Again, the best (2.61, *exploration*) and worst (3.13, *script*) results were achieved by groups in direct competition. Despite this poor starting point for the *script*-group, it outperformed the other settings (see Section 10.3.2).
- *Follow-up knowledge test.* The measure for objective knowledge gain reveals that the group β -version with negative attribution got the poorest results with 4.95 points on a scale from 0 to 9, while the group *script* performed best with an average of 6.91 points. See also Section 10.3.2.

10.3.2 Analysis of Variance

In this section, we discuss significant inter-cell dependencies of the different test settings and detail the results according to the two hypotheses formulated above.

10.3.2.1 Hypothesis $H_{1,0}$: There is no difference in the objective knowledge the students gain from a traditional lecture or from computer-based learning

In the 'outside world', computer-based learning generally comprises either no guidance at all, or it contains a guided tour at the beginning. In other words, *exploration* is the usual approach to computer-based training. Some programs have incorporated a mechanism of feedback, where the learner might access a survey of multiple choice questions, etc. This feedback mechanism depicts an intense pre-occupation with the learning program. In our test setting, the *script* plays this part of a didactically elaborated use of the modules.

In the evaluation of traditional learning versus computer-based learning, therefore, we have concentrated on the three settings *lecture*, *script*, and *exploration*, as discussed above. Table 10.3 presents the test results for the significance and for the *explained variance*

$$\eta^2 = \frac{QS_{\text{treat}}}{QS_{\text{total}}},$$

of inter-cell dependencies, where QS_{treat} is the treatment square sum and QS_{total} is the square sum of the total distribution [Bor93]. The significance p indicates the error probability of wrongly discarding the hypothesis $H_{1,0}$. When this error probability is less than 5%, the correlation is called *significant*.

Source	Dependent variable	Sig. p	η^2
$(H_{1,0}^{A1}) - \text{Lecture, Script, Exploration}$			
Preliminary knowl. test	Follow-up knowl. test	0.001**	0.163
Preliminary knowl. test	Mean rating	0.938	0.000
Setting	Follow-up knowl. test	0.024*	0.104
Setting	Mean rating	0.580	0.016
$(H_{1,0}^{B1}) - \text{Lecture, Exploration}$			
Preliminary knowl. test	Follow-up knowl. test	0.027*	0.099
Preliminary knowl. test	Mean rating	0.493	0.010
Setting	Follow-up knowl. test	0.222	0.032
Setting	Mean rating	0.505	0.010
$(H_{1,0}^{C1}) - \text{Lecture, Script}$			
Preliminary knowl. test	Follow-up knowl. test	0.005**	0.155
Preliminary knowl. test	Mean rating	0.971	0.000
Setting	Follow-up knowl. test	0.098	0.056
Setting	Mean rating	0.268	0.025
$(H_{1,0}^{D1}) - \text{Exploration, Script}$			
Preliminary knowl. test	Follow-up knowl. test	0.000***	0.296
Preliminary knowl. test	Mean rating	0.593	0.007
Setting	Follow-up knowl. test	0.003**	0.200
Setting	Mean rating	0.846	0.001

Table 10.3: Test of the significance p and explained variance η^2 of inter-cell dependencies for hypothesis $H_{1,0}$. The significant dependencies are highlighted ($p < 0.05 = *$, $p < 0.01 = **$, $p < 0.001 = ***$).

We were especially interested in the influence of the preliminary knowledge test as well as of the learning setting (i.e., *lecture*, *script*, *exploration*) on the two variables: follow-up knowledge test and subjective rating of the applets. The results in Table 10.3 were calculated by methods of covariance analysis.

As can be seen from Table 10.3 ($H_{1,0}^{A1}$), the a-priori knowledge of the students has a highly significant influence (i.e., $p = 0.001$) on the follow-up knowledge test. Even more, the preliminary knowledge explains 16.3% of the follow-up test results. On the other hand, the students' preliminary knowledge has absolutely no influence (i.e., $p = 0.938$) on their subjective rating of the applets. The learning

setting also significantly (i.e., $p = 0.024$) influences the students' knowledge gain, and this explains another 10.4% of the test results on the follow-up test.

Analogously, the interpretation of the influence can be regarded when each *two* pairs of settings are taken into consideration. In Table 10.3 ($H_{1;0}^{B1}$), ($H_{1;0}^{C1}$) and ($H_{1;0}^{D1}$), each pair of learning scenarios is evaluated. As a result, Table 10.3 shows that the a-priori knowledge of the students *always* significantly influences the follow-up knowledge test but that the subjective rating of the program is not influenced. However, in ($H_{1;0}^{B1}$) and ($H_{1;0}^{C1}$) the setting has no significance on the follow-up knowledge test, in contrast to ($H_{1;0}^{A1}$). Thus, there must be a significance between the settings: *exploration* and *script* (as the only remainders). Luckily, ($H_{1;0}^{D1}$) numerically supports this statement: The setting of either *exploration* or *script* has a highly significant influence on the follow-up knowledge test.

Dependent Variable	Setting	mean \bar{x}	std. dev.	95% Confidence Interval	
				lower border	upper border
($H_{1;0}^{A2}$) – Lecture, Script, Exploration					
Mean rating	<i>Lecture</i>	2.317	0.102	2.114	2.520
Mean rating	<i>Script</i>	2.469	0.117	2.237	2.702
Mean rating	<i>Exploration</i>	2.436	0.119	2.199	2.674
Follow-up knowl. test	<i>Lecture</i>	6.253	0.285	5.685	6.822
Follow-up knowl. test	<i>Script</i>	6.999	0.326	6.348	7.651
Follow-up knowl. test	<i>Exploration</i>	5.698	0.333	5.033	6.364
($H_{1;0}^{B2}$) – Lecture, Exploration					
Mean rating	<i>Lecture</i>	2.322	0.103	2.114	2.529
Mean rating	<i>Exploration</i>	2.429	0.121	2.185	2.673
Follow-up knowl. test	<i>Lecture</i>	6.308	0.308	5.689	6.927
Follow-up knowl. test	<i>Exploration</i>	5.718	0.362	4.989	6.446
($H_{1;0}^{C2}$) – Lecture, Script					
Mean rating	<i>Lecture</i>	2.316	0.090	2.135	2.497
Mean rating	<i>Script</i>	2.470	0.103	2.262	2.678
Follow-up knowl. test	<i>Lecture</i>	6.282	0.283	5.713	6.852
Follow-up knowl. test	<i>Script</i>	7.014	0.325	6.360	7.669
($H_{1;0}^{D2}$) – Exploration, Script					
Mean rating	<i>Script</i>	2.471	0.129	2.211	2.731
Mean rating	<i>Exploration</i>	2.435	0.132	2.169	2.701
Follow-up knowl. test	<i>Script</i>	6.933	0.290	6.347	7.520
Follow-up knowl. test	<i>Exploration</i>	5.617	0.297	5.017	6.218

Table 10.4: Estimated mean values, standard deviation and confidence intervals of the dependent variable at the different learning settings for hypothesis $H_{1;0}$ when the values of Table 10.3 are taken into consideration.

When the influence of the covariate preliminary knowledge test is taken into account, Table 10.4 gives a reliable estimate of the follow-up knowledge test and the mean rating.

The result in ($H_{1;0}^{A2}$) is that the computer-based setting *script* allows to anticipate the best results in the knowledge test while the *exploration* setting performs worst, and this difference is significant as shown in Table 10.3. Note that the knowledge test contains nine questions, thus nine possible points

(see Section A.2.2). An expected result of 5.698 to 6.999 for the follow-up test is thus very high in either setting. However, the mean program rating remains relatively constant. Since the maximal rating was three, and the expected rating in each setting is between 2.317 and 2.469, we already encounter a *floor effect*, i.e., the rating is so good that a normal distribution is no longer possible.

In $(H_{1;0}^{B2})$ and $(H_{1;0}^{C2})$, the expected values for the follow-up knowledge test are quite comparable. In $(H_{1;0}^{B2})$, the *lecture* setting slightly wins the bid, while in $(H_{1;0}^{C2})$, *script* slightly outperforms *lecture*. Neither of these differences is significant, though (see Table 10.3). A strong difference, however, can be observed in $(H_{1;0}^{D2})$: The setting *script* outperforms the setting *exploration* by a difference of 1.316 on the results of the knowledge test, and this difference is highly significant (see Table 10.3).

Summary and Conclusion

The dependencies and expected results in the different learning settings that we have deduced in this section allow comparable interpretations. The lecture held by Prof. Effelsberg yields a knowledge gain comparable to that in a computer-based learning scenario. However, we have proven that the results depend on the precise setting. This is the reason why the literature supports both statements: that a lecture is superior to good computer-based training and vice versa. Of the three settings *lecture*, *exploration*, and *script*, the last one yielded the highest scores since the script directed the attention of the students to the program at different stages, and in different contexts. This result is especially noteworthy since the lecture of Prof. Effelsberg was rated extremely positively by the students (see Section 10.2.1.1). But in contrast to all other learning settings, we observed our students in the setting *script* fetching paper and a pencil to take notes as they studied. In contrast to a guided tour at the beginning of a program, the *script* has the additional advantage of capturing attention in between, not just at the beginning.

10.3.2.2 Hypothesis $H_{2;0}$: There is no difference in the objective knowledge the students gain from different settings of computer-based learning

The variance test of this second hypothesis was especially thrilling since all participating students encountered the *identical* learning information: a 20-minute video plus two Java applets on the one- and two-dimensional discrete cosine transforms (see Section 10.2.1). However, the attributive information on the background of the programs varied (see Section A.1 for the original documents). Table 10.5 shows the results of the variance analysis on the influence of the two variables *preliminary knowledge test* and *learning setting* on the two variables *follow-up knowledge test* and *subjective rating* of the applets.

It becomes immediately obvious that much more significant correlations were observed than in the previous test. In Table 10.5 ($H_{2;0}^{A1}$), we can see that the preliminary knowledge test again significantly influences the results of the follow-up test. Moreover, it explains 24.7% of the follow-up test, which is much stronger than the 16.3% influence that we measured in Table 10.3. This means, we were a bit ‘unlucky’ with the distribution of the students over the settings since there was a significant difference in preliminary knowledge between the settings. Table 10.2 indeed indicates that the test group *c’t-article* was especially poor and the test group *β -version* was especially strong in the preliminary knowledge test. However, the *setting* in $(H_{2;0}^{A1})$ has a highly significant influence on the follow-up test results, and this explains 17.0% of the actual results, which is much stronger than the counterpart of

Source	Dependent variable	Sig. p	η^2
$(H_{2;0}^{A1}) - \text{Exploration, } \beta\text{-version, } c't\text{-article}$			
Preliminary knowl. test	Follow-up knowl. test	0.000***	0.247
Preliminary knowl. test	Mean rating	0.414	0.012
Setting	Follow-up knowl. test	0.004**	0.170
Setting	Mean rating	0.027*	0.118
$(H_{2;0}^{B1}) - \text{Exploration, } \beta\text{-version}$			
Preliminary knowl. test	Follow-up knowl. test	0.000***	0.415
Preliminary knowl. test	Mean rating	0.828	0.001
Setting	Follow-up knowl. test	0.047*	0.098
Setting	Mean rating	0.043*	0.101
$(H_{2;0}^{C1}) - \text{Exploration, } c't\text{-article}$			
Preliminary knowl. test	Follow-up knowl. test	0.066	0.086
Preliminary knowl. test	Mean rating	0.365	0.022
Setting	Follow-up knowl. test	0.157	0.052
Setting	Mean rating	0.714	0.004
$(H_{2;0}^{D1}) - \beta\text{-version, } c't\text{-article}$			
Preliminary knowl. test	Follow-up knowl. test	0.001**	0.274
Preliminary knowl. test	Mean rating	0.328	0.025
Setting	Follow-up knowl. test	0.001**	0.236
Setting	Mean rating	0.005**	0.193

Table 10.5: Test of the significance p and explained variance η^2 of inter-cell dependencies for hypothesis $H_{2;0}$. The significant dependencies are highlighted ($p < 0.05 = *$, $p < 0.01 = **$, $p < 0.001 = ***$).

10.4% in Table 10.3. Moreover, the setting significantly influences the average subjective rating of the program.

The interpretation of $(H_{2;0}^{B1})$ is analogous. The setting significantly influences both the follow-up knowledge test and the program rating. In hypothesis $H_{1;0}$, neither of the two dependencies were observed. This means that the affiliation of the students within one of the two groups *exploration* or *β -version* is of utmost importance for their subjective rating as well as for their objective knowledge gain. Note that the difference between both settings encompasses *one single sentence*: ‘Diese Lernmodule basieren auf einer Studienarbeit, die nachträglich ergänzt und erweitert wurde’, i.e., ‘These learning modules are based on a student’s implementation which subsequently has been upgraded and enlarged’ (see Sections A.1.1 and A.1.3). This sole notion that the presented applets have been implemented as a *β -version* makes all the difference of $(H_{2;0}^{B1})$!

The interpretations of the cells $(H_{2;0}^{C1})$ and $(H_{2;0}^{D1})$ are straightforward. Since it is the setting *β -version* which provokes the strong differences, $(H_{2;0}^{C1})$ does not show any significant dependencies, while $(H_{2;0}^{D1})$ proves an even stronger correlation between the setting and the two parameters of interest: follow-up knowledge test and average rating. The *$c't$ -article* was valued so highly by the students that the setting (i.e., *β -version* versus *$c't$ -article*) explains 23.6% of the results in the follow-up knowledge test, which is much stronger than the already strong influence of 9.8% in $(H_{2;0}^{B1})$.

Dependent Variable	Setting	mean \bar{x}	std. dev.	95% Confidence Interval	
				lower border	upper border
$(H_{2;0}^{A2}) - \text{Exploration, } \beta\text{-version, } c't\text{-article}$					
Mean rating	<i>Exploration</i>	2.439	0.118	2.203	2.674
Mean rating	$\beta\text{-version}$	2.075	0.118	1.838	2.313
Mean rating	<i>c't-article</i>	2.518	0.122	2.275	2.762
Follow-up knowl. test	<i>Exploration</i>	5.609	0.340	4.927	6.290
Follow-up knowl. test	$\beta\text{-version}$	4.768	0.343	4.082	5.454
Follow-up knowl. test	<i>c't-article</i>	6.480	0.353	5.774	7.185
$(H_{2;0}^{B2}) - \text{Exploration, } \beta\text{-version}$					
Mean rating	<i>Exploration</i>	2.435	0.124	2.183	2.687
Mean rating	$\beta\text{-version}$	2.065	0.124	1.813	2.317
Follow-up knowl. test	<i>Exploration</i>	5.736	0.301	5.127	6.345
Follow-up knowl. test	$\beta\text{-version}$	4.859	0.301	4.250	5.468
$(H_{2;0}^{C2}) - \text{Exploration, } c't\text{-article}$					
Mean rating	<i>Exploration</i>	2.450	0.129	2.189	2.711
Mean rating	<i>c't-article</i>	2.519	0.132	2.251	2.787
Follow-up knowl. test	<i>Exploration</i>	5.561	0.372	4.808	6.315
Follow-up knowl. test	<i>c't-article</i>	6.336	0.382	5.563	7.108
$(H_{2;0}^{D2}) - \beta\text{-version, } c't\text{-article}$					
Mean rating	$\beta\text{-version}$	2.079	0.100	1.876	2.282
Mean rating	<i>c't-article</i>	2.517	0.103	2.309	2.725
Follow-up knowl. test	$\beta\text{-version}$	4.744	0.347	4.043	5.446
Follow-up knowl. test	<i>c't-article</i>	6.468	0.355	5.749	7.188

Table 10.6: Estimated mean values, standard deviation and confidence intervals of the dependent variable at the different learning settings for hypothesis $H_{2;0}$ when the values of Table 10.5 are taken into consideration.

Table 10.6 shows the estimate of the outcomes of the follow-up knowledge test and the average rating when the influence of the covariate preliminary knowledge test has been purged. When the setting *exploration* is again taken as the reference group, Table 10.6 ($H_{2;0}^{A2}$) clearly states that a negative attribution to a program (i.e., $\beta\text{-version}$) whittles down both the subjective rating and the objective knowledge gain. Conversely, a positive attribution (i.e., *c't-article*) increases both. These differences are significant (see Table 10.5). The fact that the loss for the negative attribution is much stronger than the gain for the positive attribution can again be explained by the floor effect: The results are already so good (i.e., 2.075, 2.439, and 2.518 on a scale with a minimum of 0 and a maximum of 3) that an even higher score would not allow a normal distribution. The results in the cells ($H_{2;0}^{B2}$), ($H_{2;0}^{C2}$), and ($H_{2;0}^{D2}$) are comparable to ($H_{2;0}^{A2}$) although the exact numbers vary slightly due to the different backgrounds.

Summary and Conclusion

We have proven that the hypothesis $H_{2;0}$ on the comparability of different settings of computer-based learning must be discarded with high significance. A single sentence indicating that the simulation applets were developed by a student lowers results dramatically. Inversely, a positive attribution of the

programs produces better results, though with a percentage of 6.6% (see Table 10.5 ($H_{2;0}^{C1}$)) they fall just short of significance. What is more, not only the *subjective rating* of the programs is influenced by this attribution but the *objective gain of knowledge* as well, which decreases with negative attribution (see $H_{2;0}^{B2}$), while positive attribution increases it. The total difference in knowledge gain is enormous at 36.36% (see $H_{2;0}^{D2}$). A common practice of universities is to distribute software labeled as ‘own development’. Our evaluation indicates clear a need for change: *Never say β !*

Chapter 11

Conclusion and Outlook

*‘Where shall I begin, please your Majesty?’
he asked. ‘Begin at the beginning,’ the King
said, gravely, ‘and go on till you come to the
end: then stop.’*

– Lewis Carroll

This dissertation encompasses two major points of discussion. Firstly, it investigates possible applications of the wavelet transform in the multimedia environment. That is, in the fields of audio, still images, and video coding. In a second focal point, it re-considers mathematical transformations and related schemes in the general context of teaching.

The development of wavelet-based multimedia tools currently is an active field of research. Motivated by the research environment on multimedia at the Department Praktische Informatik IV, we were interested in promising novel wavelet-based applications for analysis and compression. The tools that were developed in the framework of this thesis are quite general and may prove useful in a variety of audio, image and video processing applications. However, only a small number could be investigated within its scope. Many improvements and extensions can be envisaged.

For example, we have restricted our investigation of digital audio coding by means of the wavelet transform to denoising a signal disturbed by white and Gaussian noise. Our audio denoising tool is the first software to underline a theoretical discussion on wavelet-based denoising. It does yet not allow a direct comparison to other denoising approaches. This surely is an open issue for further research.

The new coding standard JPEG2000 is based on the wavelet transform, implemented via the lifting scheme. Our investigation of wavelet-based still image coding, however, rests upon the convolution-based filter implementation, as it allows more flexibility in the choice of parameter settings. Our contribution to still image coding was to use the multiscale property of the wavelet transform to successfully extract the semantic feature of edges from still images. This idea resulted in a novel algorithm for semiautomatic image segmentation. It will have to be further sounded out and refined in order to obtain a stable approach for different classes of images and objects. Furthermore, we have evaluated a best setting of the many parameters of a convolution-based wavelet implementation, where we have restricted ourselves to orthogonal compactly supported Daubechies filter banks.

Clearly, our evaluation of parameter settings could be extended in many directions. With the inclusion of different classes of wavelet filters, an even deeper comprehension of the theory would have been possible. A third investigation on still images has selected a specific topic of the JPEG2000 standard, regions-of-interest coding, and has critically discussed its strengths and weaknesses.

Our engagement in a teleteaching project, where lectures and seminars are transmitted to remote locations, has pointed out the problem of allowing participants with different access bandwidths to dial into a video session. Existing hierarchical video codecs are based on the discrete cosine transform. Our substantial contribution to hierarchical video coding was to successfully exploit the wavelet transform. We addressed this novel approach both theoretically and by implementing a hierarchical video codec. We suggested a policy for the distribution of the transformed and quantized coefficients onto the different video layers and presented a prototype for a hierarchical client-server video application.

In the evaluation of still images as well as of digital video, we were faced with the challenge of automatically and objectively assessing the quality of a distorted image or video. In general, computational models of the human visual system are still in their infancy, and many issues remain to be solved. Though a number of research groups have published attempts to measure digital distortions analogous to the subjective rating of test subjects, our own empirical subjective evaluations pointed out that the much vilified signal-to-noise ratio wrongly bears this bad reputation: Our evaluation pointed out that the PSNR correlates better to the human visual perception than many of the so-called 'intelligent' metrics.

The evaluation of the learning behavior and progress made by students learning by means of a computer-based training versus that of students in a lecture was one of the most exhaustive and thorough evaluations ever conducted in this area. It not only revealed that a good computer-based training program can outperform all knowledge gain by students in a lecture scenario, it also precisely states which circumstances, i.e., attributes have what effect. An open issue in this regard is to get away from both the introductory video and the global help systems of the computer-based setting, and to experiment with smaller units of instruction. We will try to explore which arguments in the information notes induce which precise reaction. Where is the limit of the plausibility of both positive attribution (here: c't-article) and negative attribution (here: β -version)? A logical extension of our test setting is to combine both positive attribution and script. When the students are told that they are working with a 'groovy' product and they are furthermore being aided by the sustaining element of the script, questions arise such as *is there an upper limit to what can be reached with the program, can this upper limit be met, and might a positive attribution already be enough, so that the script might be omitted without any negative effect?* These didactic-psychological evaluations will be continued at the University of Mannheim in the semesters to come.

Finally, the thorough evaluation of the Java applets stands in contrast to other evaluations conducted in the progress of the presented work. Two main reasons are responsible for this. A valid accomplishment of an evaluation requires great expertise, as we have learned by doing. Furthermore, the evaluation presented in the final part of this dissertation took six months from its initial planning until the results in terms of numbers and correlations. Since the ideas which we have presented required both a feasibility study and implementation, our limited time obliged us to economize our forces, and thus forego additional thorough evaluations.

Part IV

Appendix

Appendix A

Original Documents of the Evaluation

Es muss z.B. das Gehör mit dem Gesicht, die Sprache mit der Hand stets verbunden werden, indem man den Wissensstoff nicht bloss durch Erzählungen vorträgt, dass er in die Ohren eindringe, sondern auch bildlich darstellt, damit er sich durch das Auge der Vorstellung einpräge. Die Schüler ihrerseits sollen früh lernen, sich mit der Sprache und der Hand auszudrücken, und keine Sache soll beiseite gelegt werden, bevor sie sich dem Ohr, dem Auge, dem Verstand und dem Gedächtnis hinreichend eingeprägt hat.

– Johannes Amos Comenius

A.1 Computer-based Learning Setting

All probands of the *computer-based learning* setting were provided with an introductory paper which differed according to the underlying setting. In the following sections, the original documents are quoted.

A.1.1 Setting: *Exploration*

Liebe Studierende!

In diesem Semester werden die *Lernmodule zur Bildverarbeitung mit Hilfe der Diskreten Cosinus- und Fourier-Transformation* evaluiert. Die Lernmodule sollen zu einem verbesserten Einsatz multimedialer Lehre beitragen und perspektivisch auch im Rahmen eines Fernstudiums nutzbar sein. Dazu ist es notwendig, diese zu evaluieren. Um eine derartige Evaluation durchzuführen, sind wir auf Ihre Mitarbeit angewiesen.

Die Datenauswertung erfolgt komplett bei den Evaluationspartnern des Projektes VIROR an der Universität Mannheim. Die erhobenen Daten werden zu rein wissenschaftlichen Zwecken verwendet und streng vertraulich behandelt. Es werden keine Daten erhoben, die auf Sie als Person zurückschließen lassen.

Die Ergebnisse der Untersuchung können Sie im Wintersemester 2001/2002 unter www.viror.de finden. Um für Sie die Teilnahme interessanter zu machen, führen wir eine Verlosung verschiedener Geldpreise durch. Dazu erhalten Sie einen 'Teilnahmeschein' den Sie am 19.6.01 unbedingt mitbringen müssen, um an der Verlosung teilzunehmen.

Bitte lesen Sie sich die nachfolgende Anleitung genau durch und beachten Sie die Bearbeitungshinweise!

In den nächsten 45 Minuten sollen Sie am Laptop zwei Lehrmodule **eigenständig** bearbeiten. Wie Sie bei der Bearbeitung vorgehen, bleibt Ihnen überlassen. Bevor Sie jedoch beginnen mit den Lernmodulen zu arbeiten, beantworten Sie bitte die Fragen auf den nachfolgenden Seiten.

Bitte benutzen Sie unbedingt die Bedienungsanleitung und die Hilfeseiten in den Lernmodulen und bearbeiten Sie die Lernmodule aufmerksam und konzentriert!

Vielen Dank für Ihre Teilnahme und viel Glück bei der Preisverlosung!

A.1.2 Setting: *Script*

Liebe Studierende!

In diesem Semester werden die *Lernmodule zur Bildverarbeitung mit Hilfe der Diskreten Cosinus- und Fourier-Transformation* evaluiert. Die Lernmodule sollen zu einem verbesserten Einsatz multimedialer Lehre beitragen und perspektivisch auch im Rahmen eines Fernstudiums nutzbar sein. Dazu ist es notwendig, diese zu evaluieren. Um eine derartige Evaluation durchzuführen, sind wir auf Ihre Mitarbeit angewiesen.

Die Datenauswertung erfolgt komplett bei den Evaluationspartnern des Projektes VIROR an der Universität Mannheim. Die erhobenen Daten werden zu rein wissenschaftlichen Zwecken verwendet und streng vertraulich behandelt. Es werden keine Daten erhoben, die auf Sie als Person zurückschließen lassen.

Die Ergebnisse der Untersuchung können Sie im Wintersemester 2001/2002 unter www.viror.de finden. Um für Sie die Teilnahme interessanter zu machen, führen wir eine Verlosung verschiedener Geldpreise durch. Dazu erhalten Sie einen 'Teilnahmeschein' den Sie am 19.6.01 unbedingt mitbringen müssen, um an der Verlosung teilzunehmen.

Bitte lesen Sie sich die nachfolgende Anleitung genau durch und beachten Sie die Bearbeitungshinweise!

In den nächsten 45 Minuten sollen Sie am Laptop zwei Lehrmodule **eigenständig** bearbeiten. Wie Sie bei der Bearbeitung vorgehen, bleibt Ihnen überlassen. Bevor Sie jedoch beginnen mit den Lernmodulen zu arbeiten, beantworten Sie bitte die Fragen auf den nachfolgenden Seiten.

Bitte benutzen Sie unbedingt die Bedienungsanleitung und die Hilfeseiten in den Lernmodulen und bearbeiten Sie die Lernmodule aufmerksam und konzentriert!

Vielen Dank für Ihre Teilnahme und viel Glück bei der Preisverlosung!

Zur Bearbeitung der Lernmodule

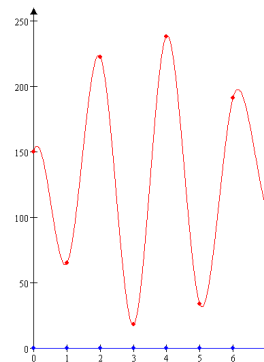
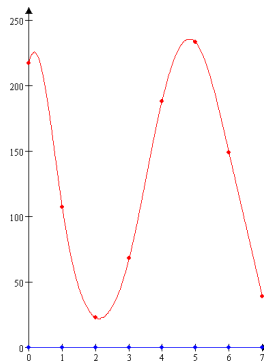
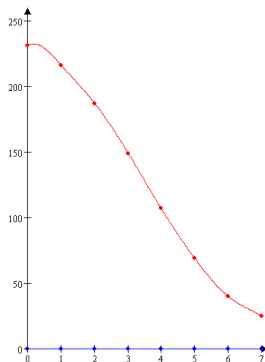
Im Rahmen der Unterrichtsforschung zeigte sich, dass verschiedene Bearbeitungsformen von Lernmodulen zu sehr unterschiedlichen Lernergebnissen führen. Um die Lernmodule möglichst effektiv zu bearbeiten, **befolgen Sie bitte die nachfolgenden Instruktionen möglichst genau.**

Es hat sich als besonders günstig erwiesen, wenn man sich bei der Bearbeitung eines Lernmoduls vorstellt, dass man die Inhalte anschließend **einer dritten Person erklären muss**. Deshalb sollten Sie sich bei der Bearbeitung **wiederholt selbst fragen**, ob Sie in der Lage sind, die zuvor bearbeiteten Inhalte einer anderen Person zu erklären/vermitteln. Die nachfolgenden Leitfragen (nächste Seite) sollen Ihnen dabei helfen zu erkennen, welche inhaltlichen Aspekte für die Erklärung der Inhalte wichtig sind. Deshalb versuchen Sie bitte, alle nachfolgenden Fragen stichpunktartig zu beantworten und sich genau zu überlegen, wie Sie dann die einzelnen Aspekte einer dritten Person erklären. Sollten Sie einzelne Fragen auch nach längerem Überlegen nicht beantworten können, dann übergehen Sie diese. Versuchen Sie aber immer, nicht nur die einzelnen Fragen zu beantworten, sondern bemühen Sie sich, einen **‘roten Faden’ in ihrer Erklärung** der einzelnen Aspekte über die verschiedenen Fragen hinweg zu entwickeln. Wie Sie aus Ihrer eigenen Lernerfahrung sicherlich wissen, können Inhalte, die in ihrem Zusammenhang erklärt werden, besser behalten werden als eine Vielzahl punktueller Fakten.

Leitfragen

Bitte denken Sie daran auch die Hilfefunktionen der Lernmodule bei der Beantwortung der Fragen zu benutzen!

1. Was vermitteln die Module?
2. Wie sind die Module aufgebaut?
3. Welche Bedeutung haben im Lernmodul eindimensionale DCT ...
 - (a) die blaue Kurve?
 - (b) die rote Kurve?
4. Was ist der Unterschied zwischen der DCT und der DFT?
5. Was wird in den acht Zahlenfeldern (unten rechts) im Lernmodul 'Eindimensionale DCT' dargestellt?
6. Was wird in den 64 Zahlenfeldern im Lernmodul 'Zweidimensionale DCT' dargestellt?
7. Wie erklärt sich der Zusammenhang zwischen Orts- und Frequenzraum?
8. Was heißt Grauwertrepräsentation?
9. Was gibt der Quantisierungsfaktor an?
10. Was meint 'Zero-Shift'?
11. Erklären Sie, wie man die nachfolgenden Abbildungen in der 1-dimensionalen DCT erzeugt.
12. Werden in den Lernmodulen Verfahren zur Datenreduktion dargestellt? Begründen Sie!



A.1.3 Setting: β -Version

Liebe Studierende!

In diesem Semester werden die *Lernmodule zur Bildverarbeitung mit Hilfe der Diskreten Cosinus- und Fourier-Transformation* evaluiert. Diese Lernmodule basieren auf einer Studienarbeit, die nachträglich ergänzt und erweitert wurde. Die Lernmodule sollen zu einem verbesserten Einsatz multimedialer Lehre beitragen und perspektivisch auch im Rahmen eines Fernstudiums nutzbar sein. Dazu ist es notwendig, diese zu evaluieren. Um eine derartige Evaluation durchzuführen, sind wir auf Ihre Mitarbeit angewiesen.

Die Datenauswertung erfolgt komplett bei den Evaluationspartnern des Projektes VIROR an der Universität Mannheim. Die erhobenen Daten werden zu rein wissenschaftlichen Zwecken verwendet und streng vertraulich behandelt. Es werden keine Daten erhoben, die auf Sie als Person zurückschließen lassen.

Die Ergebnisse der Untersuchung können Sie im Wintersemester 2001/2002 unter www.viror.de finden. Um für Sie die Teilnahme interessanter zu machen, führen wir eine Verlosung verschiedener Geldpreise durch. Dazu erhalten Sie einen 'Teilnahmeschein' den Sie am 19.6.01 unbedingt mitbringen müssen, um an der Verlosung teilzunehmen.

Bitte lesen Sie sich die nachfolgende Anleitung genau durch und beachten Sie die Bearbeitungshinweise!

In den nächsten 45 Minuten sollen Sie am Laptop zwei Lehrmodule **eigenständig** bearbeiten. Wie Sie bei der Bearbeitung vorgehen, bleibt Ihnen überlassen. Bevor Sie jedoch beginnen mit den Lernmodulen zu arbeiten, beantworten Sie bitte die Fragen auf den nachfolgenden Seiten.

Bitte benutzen Sie unbedingt die Bedienungsanleitung und die Hilfeseiten in den Lernmodulen und bearbeiten Sie die Lernmodule aufmerksam und konzentriert!

Vielen Dank für Ihre Teilnahme und viel Glück bei der Preisverlosung!

A.1.4 Setting: *c't*-Article

Liebe Studierende!

In diesem Semester werden die *Lernmodule zur Bildverarbeitung mit Hilfe der Diskreten Cosinus- und Fourier-Transformation* evaluiert. Die Lernmodule sollen zu einem verbesserten Einsatz multimedialer Lehre beitragen und perspektivisch auch im Rahmen eines Fernstudiums nutzbar sein. Dazu ist es notwendig, diese zu evaluieren. Um eine derartige Evaluation durchzuführen, sind wir auf Ihre Mitarbeit angewiesen.

Wie Sie an dem vor Ihnen liegenden Vorabauszug eines Artikels der kommenden Ausgabe der Computerfachzeitschrift *c't* sehen, wurden die zu bearbeitenden Lernmodule als vorbildliche Beispiele des zukünftigen Studierens durch Prof. Dr. L. Kämmerer (ETH Zürich) eingestuft. Prof. Kämmerer ist zudem auch regelmäßiger Autor in der *c't*. Wir möchten uns an dieser Stelle bei der *c't*-Redaktion und Herrn Prof. Dr. Kämmerer für die Möglichkeit der Vorab-Publikation (siehe Kopie auf der nächsten Seite) bedanken!

Die Datenauswertung erfolgt komplett bei den Evaluationspartnern des Projektes VIROR an der Universität Mannheim. Die erhobenen Daten werden zu rein wissenschaftlichen Zwecken verwendet und streng vertraulich behandelt. Es werden keine Daten erhoben, die auf Sie als Person zurückschließen lassen.


Die Ergebnisse der Untersuchung können Sie im Wintersemester 2001/2002 unter www.viror.de finden. Um für Sie die Teilnahme interessanter zu machen, führen wir eine Verlosung verschiedener Geldpreise durch. Dazu erhalten Sie einen 'Teilnahmeschein' den Sie am 19.6.01 unbedingt mitbringen müssen, um an der Verlosung teilzunehmen.

Bitte lesen Sie sich die nachfolgende Anleitung genau durch und beachten Sie die Bearbeitungshinweise!

In den nächsten 45 Minuten sollen Sie am Laptop zwei Lehrmodule **eigenständig** bearbeiten. Wie Sie bei der Bearbeitung vorgehen, bleibt Ihnen überlassen. Bevor Sie jedoch beginnen mit den Lernmodulen zu arbeiten, beantworten Sie bitte die Fragen auf den nachfolgenden Seiten.

Bitte benutzen Sie unbedingt die Bedienungsanleitung und die Hilfeseiten in den Lernmodulen und bearbeiten Sie die Lernmodule aufmerksam und konzentriert!

Vielen Dank für Ihre Teilnahme und viel Glück bei der Preisverlosung!



Report
Virtuelles Lernen im Studium

Skripte zum Download helfen Studenten, die eine Vorlesung verpasst haben, ebenso wie Studieninteressenten, die sich ein Bild von Inhalten und Anforderungen verschaffen wollen. Und Studenten wie Dozenten können davon lernen, wie ähnlicher Stoff anderswo aufbereitet wird. Dass Dozenten ihre Skripte freigeben, ist auch ein Beitrag zur Qualitätssicherung. Genauso wie Studenten reichlich Fehler in meinen Skripten finden, entdecke ich auch hin und wieder Macken in den Skripten von Kollegen. Das Internet bietet eine einfache Chance, die unvermeidlichen Fehler gegenseitig zu korrigieren. Dozenten, die Skripte per Kennwort blockieren oder auf den Download innerhalb der Hochschule beschränken, schaden sich im Endeffekt selbst – und ihren Studenten. Auch wer mit Tafel und Kreide unterrichtet kann seine Skripte ins Internet stellen. Halbwegs saubere handschriftliche Unterlagen, eingescannt und Seite für Seite als Bilddatei gespeichert, sind den Studenten allemal lieber, als Wort für Wort mitschreiben zu müssen. Ein perfektes Äußeres hat sowieso seine Tücken: Wer sein Skript als makelloses Buch ins Internet stellt, wird daran ungen jedes Semester größere Teile aktualisieren. Außerdem ist den Studenten mit dem knappen Tafelbild am besten gedient; langatmige Bücher finden sie auch in der Bibliothek

Fortgeschrittene Lernhilfen

Die virtuellen Hilfen der Veranstaltungen müssen nicht unbedingt auf den Skripten der Dozenten basieren.

Mathematik zum Anfassen: Durch detaillierte Simulationsmöglichkeiten erklären sich einfacher komplexe Transformationen. (http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/multimedia/2d_dct/).

Computersimulationen erklären die zentralen Verfahren der Bildverarbeitung: Das Verfahren der eindimensionalen Diskreten Cosinus Transformation lässt sich leichter durch eine entsprechende Simulation verstehen (http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/multimedia/1d_dct_and_dft/).

Weitaus stärker können Studierende durch eigens für besondere inhaltliche Probleme entwickelte Selbstlernmodule profitieren. So wurden an der Universität Mannheim (LS Effelsberg) im Rahmen des Projektes „Virtuelle Universität Oberrhein“ (VIROR) [2] Java-Animationen entwickelt, die zentrale Themen der Videokompression darstellen. In solchen Java-Aminationen kann der Benutzer interaktiv agieren, wodurch ihm ansonsten schwer verständliche, abstrakte Probleme vereinfacht verdeutlicht werden. Nun werden z.B. Applets zur ein- und zweidimensionalen DCT (Discrete Cosine Transform) und zur DFT (Discrete Fourier Transform) zusätzlich zur regulären Lehre eingesetzt. Die meisten Studierenden hatten bisher im konventionellen Unterricht größte Schwierigkeiten, diese Verfahren in ihrer Funktionsweise zu verstehen. In einer Pilotstudie an der ETH

Zürich [3] führten diese ergänzenden Lehrangebote zu einer drastischen Verbesserung des Verständnisses der Studenten zu diesem Thema und wurden sehr positiv angenommen. Langfristig werden derartige beispielhafte Lernmodule sicherlich die Studienmaterialien stark verbessern.

Kontrolle ist besser

Hier und dort können die Studenten Lösungen, Laborberichte und Hausarbeiten elektronisch einreichen. Jedoch drohen dabei die bekannten Rechtsprobleme – etwa, wenn ein Student behauptet, er habe einen Bericht rechtzeitig abgesandt, der müsse wohl unterwegs verschüttet gegangen sein.

Laut deutscher Kultusministerkonferenz können Scheine für virtuelle Laborübungen und Praktika von einem Kolloquium abhängig gemacht werden per Rechner erbrachte Leistungen dürfen nur in Verbindung mit einem herkömmlichen Prüfungsgespräch bewertet werden [4]. Zumindest der Weg zu Online-Übungen, -Laboren und -Praktika ist damit frei. Beispiele für Letztere findet man aber noch selten. Online-Übungen lassen sich dagegen mit simpler Technik einrichten. So können nicht nur Studenten der Fernuniversität Hagen Rechenergebnisse oder die Nummer von Multiple-Choice-Lösungen in Web-Formulare eintippen. Der Vorteil solcher Lösungen besteht vor allem in der maschinellen Auswertung: Die Studenten erhalten sofort eine Rückmeldung, und zu jeder Vorlesungsstunde lässt sich ein Dutzend Aufgaben stellen, ohne dass Tutoren in Korrekturarbeit ersticken. Die Studenten prüfen sich sozusagen selbst: 'self-assessment'. Solche Selbsttests könnten auch Probeklausuren ersetzen oder Studieninteressenten helfen, vorab Wissen und Fähigkeiten zu testen. Zudem verbessern sie durch die kontinuierliche Selbsttestung das Leistungsniveau der Studierenden.

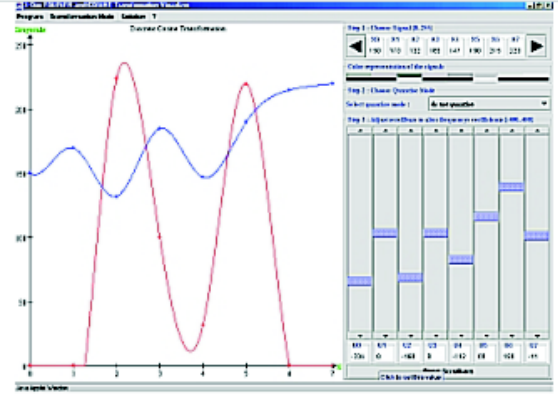
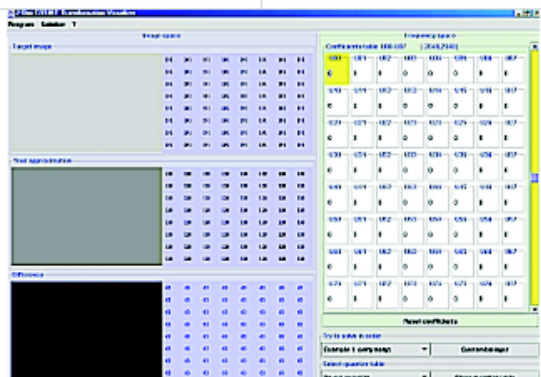



Figure A.1: c't-Article.

A.2 Knowledge Tests

A.2.1 Preliminary Test

Liebe Studierende,

Die nachfolgenden Fragebögen dienen der Erfassung zentraler Aspekte (Vorwissen, Stimmung, etc.), die den Lernprozess beeinflussen. Ihre Antworten werden selbstverständlich anonym erhoben und ausschließlich zu wissenschaftlichen Zwecken ausgewertet. Bitte tragen sie nur die Ihnen zugewiesene Nummer oben links auf alle von Ihnen bearbeitete Bögen ein.

Vielen Dank für Ihre Mitarbeit!

Zunächst ein paar Fragen zu Ihrer Person:

1. Geschlecht: männlich ☐ weiblich ☐
2. Alter: _____ Jahre
3. Studienfach: _____ Semester: _____
4. Seit wie vielen Jahren nutzen Sie bereits einen Computer?
 - (a) Studium/Schule/Beruf: _____ Jahre
 - (b) privat: _____ Jahre
5. Wieviel Zeit in Stunden verbringen Sie durchschnittlich pro Woche mit dem Computer?
 - (a) Studium/Beruf: _____ Stunden
 - (b) privat: _____ Stunden

Instruktion:

Im folgenden finden Sie eine **Liste von Wörtern, die verschiedene Stimmungen beschreiben**. Bitte gehen Sie die Wörter der Liste nacheinander durch und kreuzen Sie bei **jedem Wort** das Kästchen an, das die **augenblickliche** Stärke Ihrer Stimmung am besten beschreibt.

Im Moment fühle ich mich ...

	überhaupt nicht					sehr
1. zufrieden	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
2. ausgeruht	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
3. ruhelos	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
4. schlecht	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
5. schlapp	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
6. gelassen	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
7. müde	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
8. gut	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
9. unruhig	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
10. munter	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
11. unwohl	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	
12. entspannt	<input type="checkbox"/> ₁	<input type="checkbox"/> ₂	<input type="checkbox"/> ₃	<input type="checkbox"/> ₄	<input type="checkbox"/> ₅	

Vorkenntnisse

1. Welche Note haben Sie in der Klausur 'Praktische Informatik 1' erzielt?
Note: ____ ☐ Ich habe diese Klausur nicht mitgeschrieben.
2. Welche Noten haben Sie in der Klausur 'Lineare Algebra 1' erzielt?
Note: ____ ☐ Ich habe diese Klausur nicht mitgeschrieben.
3. Welche Noten haben Sie in der Klausur 'Analysis 1' erzielt?
Note: ____ ☐ Ich habe diese Klausur nicht mitgeschrieben.

Bei den nachfolgenden Fragen ist immer nur eine Antwort richtig:

1. Gerade Parität bedeutet, dass
 - (a) die Summe aller gesetzten Bits in einem Codewort gerade ist.
 - (b) die Anzahl der Paritätsbits gerade ist.
 - (c) es eine gerade Anzahl gültiger Codeworte gibt.
2. Der Wert -64 ist mit Hilfe des Einerkomplements und 7 Bits
 - (a) 11111112
 - (b) 10000002
 - (c) nicht darstellbar
3. Bei n Bits und Zweierkomplementdarstellung ist die kleinste darstellbare Zahl
 - (a) -2^{n-1}
 - (b) $-2^n - 1$
 - (c) -2^n
4. Ein Flipflop ...
 - (a) ... ist eine Schaltung mit einem stabilen Zustand.
 - (b) ... ist eine Schaltung mit zwei stabilen Zuständen.
 - (c) ... wechselt ständig zwischen zwei Zuständen hin und her.
5. Die Hamming-Distanz eines Codes lässt sich bestimmen durch ...
 - (a) ... die Hamming-Distanz des längsten und des kürzesten Wortes des Codes.
 - (b) ... die Hamming-Distanz zweier beliebiger Codewörter.
 - (c) ... Vergleich der Hamming-Distanzen aller Paare von Codewörtern.
6. Warum werden Daten komprimiert?
 - (a) für bessere Übertragungszeiten in Internet.
 - (b) um graphisch qualitativ bessere Grafiken im Internet darzustellen.
 - (c) weil sie sonst nicht zu speichern sind.
7. Was ist das Grundprinzip von verlustbehafteter Kompression? Es werden Daten verworfen, ...
 - (a) die in ihrer digitalen Darstellung zu viele Bits in Anspruch nehmen.
 - (b) die Nahe bei Null liegen.
 - (c) deren Einfluss auf die menschl. Wahrnehmung am Geringsten ist.
 - (d) die in ihrem digitalen Datenträger am Ende stehen.

A.2.2 Follow-up Test

Liebe Studierende,

Die nachfolgenden Fragen sollen Ihre aktuelle Stimmung, für den Lernprozess relevante Selbstbewertungen, ihre Meinung zum Lernmodul und zum Schluss Ihr erworbenes Wissen erfassen. Bitte bedenken Sie, dass es **besonders wichtig** für uns ist, dass Sie alle Fragen **allein** bearbeiten.

Vielen Dank für Ihre Mitarbeit!

Instruktion:

Im folgenden finden Sie eine **Liste von Wörtern, die verschiedene Stimmungen beschreiben**. Bitte gehen Sie die Wörter der Liste nacheinander durch und kreuzen Sie bei **jedem Wort** das Kästchen an, das die **augenblickliche** Stärke Ihrer Stimmung am besten beschreibt.

Im Moment fühle ich mich ...

	überhaupt nicht					sehr
1. schläfrig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2. wohl	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3. ausgeglichen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4. unglücklich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
5. wach	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
6. unzufrieden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
7. angespannt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
8. frisch	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
9. glücklich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
10. nervös	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
11. ermattet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
12. ruhig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Instruktion:

Im folgenden möchten wir mehr darüber erfahren, wie Sie sich bezüglich Ihrer Leistungen zur Zeit selbst einschätzen.

1. Über meine Erfolge freue ich mich immer sehr.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
2. Über gute Leistungen freue ich mich meist nicht so sehr wie andere.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
3. Wenn mir mal etwas nicht gelingt, will ich es gleich noch einmal probieren.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
4. Wenn ich eine schlechte Leistung gezeigt habe, dann möchte ich mich am liebsten verkriechen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
5. Wenn ich lange auf etwas hingearbeitet habe und es endlich geschafft ist, fühle ich mich eher leer als dass ich mich über meine Leistung freue.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
6. Wenn ich eine schlechte Leistung erbringe, denke ich gleich darüber nach, wie ich es beim nächsten Mal besser machen kann.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
7. Wenn ich eine schlechte Leistungsbeurteilung erhalte, bin ich enttäuscht und traurig.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
8. Wenn ich etwas erreiche, das ich mir vorgenommen habe, dann bin ich richtig stolz auf mich.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
9. Wenn mir die ersten Schritte misslingen, so lasse ich mich davon nicht entmutigen, sondern strebe mein Ziel auch weiterhin entschlossen an.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
10. Wenn ich etwas Neues nicht gleich verstehe, bin ich schnell unzufrieden.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
11. Wenn ich lange Zeit konzentriert auf eine Sache hin gearbeitet habe und es dann endlich geschafft ist, bin ich ausgesprochen zufrieden mit mir.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
12. Nach einer guten Leistung denke ich eher an kommende Probleme, als dass ich mir Zeit nehme, mich zu freuen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
13. Selbst wenn ich bei nicht so wichtigen Sachen scheitere, bin ich lange Zeit niedergeschlagen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
14. Auch wenn ich nur ein kleines Problem löse, kann ich mich darüber freuen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
15. Erfolge geben mir wenig Sicherheit für zukünftige Herausforderungen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
16. Selbst wenn ich bei einer für mich wichtigen Sache versage, so glaube ich dennoch an mein Können.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
17. Konnte ich mir mein Können beweisen, dann bin ich sehr zufrieden mit mir.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
18. Selbst wenn ich ein schwieriges Problem gut bewältigt habe, bin ich nie wirklich euphorisch.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
19. Von Misserfolgen lasse ich mich nicht aus der Bahn werfen.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
20. Auch kleinere Misserfolge belasten mich oft längere Zeit.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu

Instruktion:

Bitte beurteilen Sie die Lernmodule, mit denen Sie hier gearbeitet haben, insgesamt anhand der nachfolgenden Aussagen.

1. Mit meinem hier erzielten Lernergebnis bin ich zufrieden.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
2. Ich hatte kaum Einfluss auf meinen Lernerfolg.	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
3. Ich habe den Eindruck, dass ich selbst für mein Lernergebnis mit den Lernmodulen verantwortlich bin. ¹	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
4. Die inhaltliche Aufbau der Lernmodule ist verständlich. ²	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
5. Die in den Lernmodulen installierten Hilfen sind ausreichend. ³	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
6. Ich denke, dass die Lernmodule das Lernen der betreffenden Inhalte erleichtern. ⁴	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
7. Ich empfand das Lernen mit den Lernmodulen als sinnvoll. ⁵	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
8. Ich würde die Lernmodule anderen Studierenden zum Selbststudium empfehlen. ⁶	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
9. Insgesamt finde ich die Lernmodule gelungen. ⁷	trifft nicht zu	trifft eher nicht zu	trifft eher zu	trifft zu
10. Folgendes sollte an den Lernmodulen verbessert werden: ⁸				
11. Was ich sonst noch anmerken möchte:				

The questions marked ¹ to ⁸ were adapted in the *Lecture* setting.

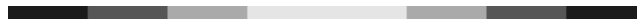
Instruktion:

Bitte beantworten Sie die nachfolgenden Fragen. Arbeiten Sie unbedingt ohne fremde Hilfe.

1. In der Bildkompression will man nach Möglichkeit die menschliche visuelle Wahrnehmung nachbilden. Wenn Sie sich ein Bild bis auf die allernotwendigste Information reduziert vorstellen (d.h. als s/w Binärbild), welche Information bleibt dann noch übrig? Um dieses in die Tat umzusetzen, malen Sie bitte das folgende Bild nach (es geht nur um das Verständnis, wir sind kein Kunstverein!):



2. Wodurch zeichnen sich Kanten aus?
3. Was für einen Sinn könnte also eine Transformation in den Frequenzraum haben?
4. Ist eine Transformation schon eine Kompression? Begründung?
5. Was bezeichnet eine Basis-Frequenz?
6. Wie oft schwingt ein \cos über dem folgenden Signal?



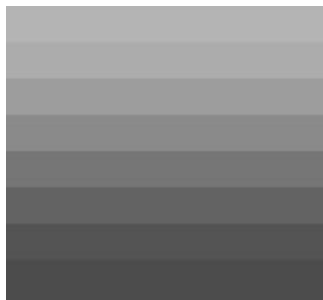
7. Was passiert, wenn man das 1-dimensionale Signal aus Aufgabe 6 auf die zweite Dimension ausdehnt, d.h., wenn man folgendes Signal transformiert?



8. Wir bezeichnen die Stärke einer Frequenzänderung als Amplitude. Wie verhalten sich die Amplituden der drei Signale zueinander?



- (a) ☐ $A > B > C$
 (b) ☐ $A > C > B$
 (c) ☐ $B > A > C$
 (d) ☐ $B > C > A$
 (e) ☐ $C > A > B$
 (f) ☐ $C > B > A$
9. Welche Einstellung der Amplituden korrespondiert zu welchem Bild?
 Verbinden Sie die entsprechenden Bilder mit den Tabellen mit einem Pfeil.



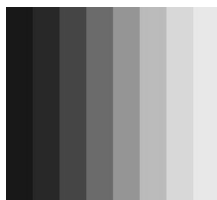
0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



0	700	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

★ Withdrawn question:

Was ändert sich, wenn das Bild gedreht wird?



A.2.3 Sample Solutions

Sample Solution of Preliminary Knowledge Test

1. Gerade Parität bedeutet, dass
 - (a) die Summe aller gesetzten Bits in einem Codewort gerade ist.
2. Der Wert -64 ist mit Hilfe des Einerkomplements und 7 Bits
 - (c) nicht darstellbar
3. Bei n Bits und Zweierkomplementdarstellung ist die kleinste darstellbare Zahl
 - (a) -2^{n-1}
4. Ein Flipflop ...
 - (b) ... ist eine Schaltung mit zwei stabilen Zuständen.
5. Die Hamming-Distanz eines Codes lässt sich bestimmen durch ...
 - (c) ... Vergleich der Hamming-Distanzen aller Paare von Codewörtern.
6. Warum werden Daten komprimiert?
 - (a) für bessere Übertragungszeiten in Internet.
7. Was ist das Grundprinzip von verlustbehafteter Kompression? Es werden Daten verworfen, ...
 - (c) deren Einfluss auf die menschl. Wahrnehmung am Geringsten ist.

Sample Solution of Follow-up Knowledge Test

1. Bild mit deutlichen Kanten für Kopf, Brille, etc.
2. Scharfe Übergänge zwischen relativ homogenen Farbbereichen.
3. Kanten zu finden.
4. Nein. Nur Basiswechsel, andere Darstellung.
5. Eine zu Verfügung stehende Frequenz, um das Ursprungssignal anzunähern.
6. einmal
7. Man braucht eine zweite Dimension. Die Koeffizienten in dieser neuen Dimension sind im Beispiel jedoch alle 0.
8. $C > B > A$
9. Linke Tabelle zum linken Bild, rechte Tabelle zum rechten Bild.

A.3 Quotations of the Students

The following quotations have been found in the follow-up test (see Section A.2.2), question 11: *What I generally want to state*. The quotations are given in their original language German.

- Effelsberg wirkte sehr motiviert, daher kam auch der Stoff, den er vermitteln wollte, wirklich gut an.
- Bin auf die weitere Entwicklung des Projektes gespannt.
- Habe mich im Praktikum intensiv mit Computer-Based Training beschäftigt und halte persönlich eher wenig davon. Im Vergleich finde ich Ihr Programm relativ gelungen.
- War echt interessant.
- Beispiele gut gelungen.
- Gut aufbereiteter Stoff.
- So etwas sollte in der Universität eingeführt werden, da es den Lernstoff verständlich erklärt.
- Es macht wirklich Spass, mit dem Modul zu arbeiten. Im spielerischen Umgang lernt man die Funktionsweise kennen und sie anzuwenden [...] ich möchte zu dem wirklich tollen Programm gratulieren.
- Tolle Sache.
- Das Einleitungsvideo fand ich sehr gut. Viel besser als Texthilfe!
- Hat Spass gemacht, allerdings ist der Zeitdruck unangenehm.

Bibliography

- [Abo99] Gregory D. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4):508–530, 1999.
- [ACM01] ACM. Computer Science Teaching Center. <http://www.cstc.org>, 2001.
- [AK99] Michael D. Adams and Faouzi Kossentini. Performance Evaluation of Reversible Integer-to-Integer Wavelet Transforms for Image Compression. In *Proc. IEEE Data Compression Conference*, page 514 ff., Snowbird, Utah, March 1999.
- [AMV96] Elan Amir, Steven McCanne, and Martin Vetterli. A Layered DCT Coder for Internet Video. In *Proc. IEEE International Conference on Image Processing*, pages 13–16, Lausanne, Switzerland, September 1996.
- [BA83] Peter Burt and Edward Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. on Communications*, COM-31(4):532–541, April 1983.
- [Bar99] Richard Baraniuk. Optimal Tree Approximation with Wavelets. In *Proc. SPIE Technical Conference on Wavelet Applications in Signal Processing*, volume 3813, Denver, July 1999.
- [BBL⁺00] Freimut Bodendorf, Christian Bauer, Christian Langenbach, Manfred Schertler, and Sascha Uelpeneich. Vorlesung auf Abruf im Internet – Lecture on Demand als Baustein einer virtuellen Universität. *Praxis der Informationsverarbeitung und Kommunikation*, 23(3):137–147, 2000.
- [Ber99] Christophe Bernard. Discrete Wavelet Analysis for Fast Optic Flow Computation. Technical report, Rapport Interne du Centre de Mathématiques Appliquées RI415, École Polytechnique, February 1999.
- [BFNS00] Katrin Borcea, Hannes Federrath, Olaf Neumann, and Alexander Schill. Entwicklung und Einsatz multimedialer Werkzeuge für die Internet-unterstützte Lehre. *Praxis der Informationsverarbeitung und Kommunikation*, 23(3):164–168, 2000.
- [BH93] Michael F. Barnsley and Lyman P. Hurd. *Fractal Image Compression*. A. K. Peters, Wellesley, MA, 1993.
- [BK97] Vasudev Bhaskaran and Konstantinos Konstantinides. *Image and Video Compression Standards*. Kluwer Academic Publishers, Norwell, MA, 1997.

- [Böm00] Florian Bömers. Wavelets in Real-Time Digital Audio Processing: Analysis and Sample Implementations. Master's thesis, Universität Mannheim, Mai 2000.
- [Boc98] Franziska Bock. *Analyse und Qualitätsbeurteilung digitaler Bilder unter Verwendung von Wavelet-Methoden*. PhD thesis, Technische Universität Darmstadt, Germany, 1998.
- [Bor93] Jürgen Bortz. *Statistik für Sozialwissenschaftler*. Springer, Berlin, Heidelberg, New York, 4th edition, 1993.
- [Bos00] Uwe Bosecker. Evaluation von Algorithmen zur Erzeugung Hierarchischer Videoströme. Master's thesis, Universität Mannheim, November 2000.
- [BS89] C. Bereiter and M. Scardamalia. Intentional learning as a goal of instruction. In L.B. Resnick, editor, *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, pages 361–392. Erlbaum, Hillsdale, NJ, 1989.
- [CAL00] Charilaos Christopoulos, Joel Askelöf, and Mathias Larson. Efficient Methods for Encoding Regions-of-interest in the Upcoming JPEG2000 Still Image Coding Standard. *IEEE Signal Processing Letters*, 7(9):247–249, September 2000.
- [CD95] Ronald R. Coifman and David L. Donoho. Translation-invariant denoising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, Lecture Notes in Statistics, pages 125–150. Springer, 1995.
- [CDDD00] Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies, and Ronald DeVore. Tree approximation and Encoding. (preprint), October 2000.
- [CEG76] A. Croisier, D. Esteban, and C. Galand. Perfect Channel Splitting by use of interpolation/decimation/tree decomposition techniques. In *Proc. International Conference on Information Sciences and Systems*, pages 443–446, Patras, Greece, August 1976.
- [Che96] Corey Cheng. Wavelet Signal Processing of Digital Audio with Applications in Electro-Acoustic Music. Master's thesis, Hanover, New Hampshire, 1996.
- [CR68] F.W. Campbell and J.G. Robson. Applications of Fourier Analysis to the Visibility of Gratings. *Journal of Physiology*, 197:551–566, 1968.
- [CS00] Elsabé Cloete and Claudia Schremmer. Addressing Problems in Virtual Learning Systems through Collaboration. In *Proc. South African Institute of Computer Scientists and Information Technologists*, Cape Town, South Africa, November 2000.
- [CYV97] G. Chang, B. Yu, and M. Vetterli. Image Denoising via Lossy Compression and Wavelet Thresholding. In *Proc. IEEE International Conference on Image Processing*, Santa Barbara, CA, October 1997.
- [Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61. SIAM. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
- [DJ89] Richard C. Dubes and Anil K. Jain. Random Field Models in Image Analysis. *Journal of Applied Statistics*, 16:131–164, 1989.

- [DJ94] David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [DJ95] David L. Donoho and Iain M. Johnstone. Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.
- [Don93a] David L. Donoho. Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data. In Daubechies, editor, *Proc. Symposia in Applied Mathematics: Different Perspectives on Wavelets*, volume 47, pages 173–205, Providence, RI, 1993.
- [Don93b] David L. Donoho. Wavelet Shrinkage and W.V.D. — A Ten Minute Tour. Technical Report 416, Stanford University, Department of Statistics, January 1993.
- [Don95] David L. Donoho. Denoising by Soft Thresholding. *IEEE Trans. on Information Theory*, 41(3):613–627, 1995.
- [DS98] Ingrid Daubechies and Wim Sweldens. Factoring Wavelet Transforms into Lifting Steps. *Journal of Fourier Analysis and Applications*, 4(3):245–267, 1998.
- [ES98] Wolfgang Effelsberg and Ralf Steinmetz. *Video Compression Techniques*. dpunkt Verlag, Heidelberg, 1998.
- [Ess01] Christoph Esser. *Studienarbeit: Wavelet-Transformation von Standbildern*. Universität Mannheim, Lehrstuhl Praktische Informatik IV, Februar 2001.
- [Füß01] Holger Füßler. JPEG2000 — Codierung von Regions-of-interest. Master’s thesis, Universität Mannheim, August 2001.
- [Fri79] John P. Frisby. *Seeing — Illusion, Brain and Mind*. Oxford University Press, Walton Street, Oxford, 1979.
- [FTWY01] T.C. Ferguson, D.M. Tan, H.R. Wu, and Z. Yu. Blocking Impairment Metric for Colour Video Images. In *Proc. International Picture Coding Symposium*, Seoul, Korea, April 2001.
- [Gao98] Hong-Ye Gao. Wavelet Shrinkage Denoising Using the Non-Negative Garrote. *Journal of Computational and Graphical Statistics*, 7(4):469–488, December 1998.
- [GB97] Hong-Ye Gao and Andrew G. Bruce. Waveshrink with firm Shrinkage. *Statistica Sinica*, 7:855–874, 1997.
- [GEE98] Werner Geyer, Andreas Eckert, and Wolfgang Effelsberg. Multimedia in der Hochschullehre: TeleTeaching an den Universitäten Mannheim und Heidelberg. In F. Scheuermann, F. Schwab, and H. Augenstein, editors, *Studieren und weiterbilden mit Multimedia: Perspektiven der Fernlehre in der wissenschaftlichen Aus- und Weiterbildung*, pages 170–196. BW Bildung und Wissenschaft Verlag und Software GmbH, Nürnberg, Germany, 1998.

- [GFBV97] Javier Garcia-Frias, Dan Benyamin, and John D. Villasenor. Rate Distortion Optimal Parameter Choice in a Wavelet Image Communication System. In *Proc. IEEE International Conference on Image Processing*, pages 25–28, Santa Barbara, CA, October 1997.
- [GGM85] P. Goupillaud, Alex Grossmann, and Jean Morlet. Cycle–octave and related transforms in seismic signal analysis. *Geoexploration*, 23:85–102, 1984/85.
- [GM85] Alex Grossmann and Jean Morlet. *Decomposition of functions into wavelets of constant shape, and related transforms*. Mathematics and Physics, Lectures on Recent Results. World Scientific Publishing, Singapore, 1985.
- [GMP85] Alex Grossmann, Jean Morlet, and T. Paul. Transforms associated to square integrable representations. I. General results. *Journal of Mathematical Physics*, 26(10):2473–2479, 1985.
- [Gol89] E. Bruce Goldstein. *Sensation and Perception*. Wadsworth Publishing Company, Belmont, CA, 1989.
- [GR98] Simon J. Godsill and Peter J.W. Rayner. *Digital Audio Restoration*. Springer, Berlin, Heidelberg, New York, 1998.
- [GW93] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Addison-Wesley, 1993.
- [Haa10] Alfréd Haar. Zur Mathematik der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.
- [Hag] Fernuniversität Hagen. WebAssign — A tool for the automation of students’ assignments. <http://www-pi3.fernuni-hagen.de/WebAssign/>.
- [Har74] Gilbert Harman. Epistemology. In *Handbook of Perception: Historical and Philosophical Roots of Perception*, pages 41–56. Academic Press, New York, 1974.
- [HBH00] Holger Horz, Andrea Buchholz, and Manfred Hofer. Neue Lehr-/Lernformen durch Teleteaching? *Praxis der Informationsverarbeitung und Kommunikation*, 23(3):129–136, 2000.
- [HDHLR99] Tia Hansen, Lone Dirckinck-Holmfeld, Robert Lewis, and Jože Rugelj. Using Telematics for Collaborative Learning. In Pierre Dillenbourg, editor, *Collaborative Learning: Cognitive and Computational Approaches*. Elsevier Science, Oxford, 1999.
- [HEMK98] Kostas Haris, Serafim N. Efstratiadis, Nicos Maglaveras, and Aggelos K. Katsaggelos. Hybrid Image Segmentation Using Watersheds and Fast Region Merging. *IEEE Trans. on Image Processing*, 7(12):1684–1699, December 1998.
- [HER⁺00] Manfred Hofer, Andreas Eckert, Peter Reimann, Nicola Döring, Holger Horz, Guido Schiffhorst, and Knut Weber. Pädagogisch–Psychologische Begleitung der ‘Virtuellen Universität Oberrhein’ VIROR (WS98/99). In Detlev Leutner and Roland Brünken, editors, *Neue Medien in Unterricht, Aus- und Weiterbildung: Aktuelle Ergebnisse empirischer pädagogischer Forschung*. Waxmann, Münster, Germany, 2000.

- [HFH01] Holger Horz, Stefan Fries, and Manfred Hofer. Stärken und Schwächen eines Teleseminars zum Thema ‘Distance Learning’. In H.M. Niegemann and K.D. Treumann, editors, *Lehren und Lernen mit interaktiven Medien (Arbeitstitel)*. Waxmann, Münster, Germany, 2001.
- [Hof97] Manfred Hofer. Lehrer–Schüler–Interaktion. In F.E. Weinert, editor, *Psychologie des Unterrichts und der Schule (Enzyklopädie der Psychologie, Themenbereich D, Serie I, Pädagogische Psychologie)*, pages 213–252. Hogrefe, Göttingen, Germany, 1997.
- [Hol95] M. Holschneider. *Wavelets: An Analysis Tool*. Oxford Science Publications, 1995.
- [Hol02] Alexander Holzinger. Hierarchische Videocodierung mit JPEG2000–codierten Einzelbildern. Master’s thesis, Universität Mannheim, Februar 2002.
- [HS85] Robert M. Haralick and Linda G. Shapiro. Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1985.
- [HSE00] Thomas Haenselmann, Claudia Schremmer, and Wolfgang Effelsberg. Wavelet–based Semi–automatic Segmentation of Image Objects. In *Proc. International Conference on Signal and Image Processing*, pages 387–392, Las Vegas, Nevada, November 2000.
- [HSKV01] Volker Hilt, Claudia Schremmer, Christoph Kuhmünch, and Jürgen Vogel. Erzeugung und Verwendung multimedialer Teachware im synchronen und asynchronen Teleteaching. *Wirtschaftsinformatik. Schwerpunkttheft ‘Virtuelle Aus- und Weiterbildung’*, 43(1):23–33, 2001.
- [Hub98] Barbara Burke Hubbard. *The world according to wavelets*. A.K. Peters, Natick, MA, 1998.
- [Irt96] Hans Irtel. *Entscheidungs– und testtheoretische Grundlagen der Psychologischen Diagnostik*. Peter Lang, Frankfurt/Main, 1996.
- [ISLG00] Frank Imhoff, Otto Spaniol, Claudia Linnhoff–Popien, and Markus Gerschhammer. Aachen–Münchener Teleteaching unter Best–Effort–Bedingungen. *Praxis der Informationsverarbeitung und Kommunikation*, 23(3):156–163, 2000.
- [ISO95] ISO/IEC 13818-2. Information technology – Generic coding of moving pictures and associated audio – Part 2: Video, 1995.
- [ITU96] ITU. *Video Coding for Low Bitrate Communication. Recommendation H.263*. International Telecommunication Union, 1996.
- [ITU00] ITU. *JPEG2000 Image Coding System. Final Committee Draft Version 1.0 – FCD15444-1*. International Telecommunication Union, March 2000.
- [Jai89] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [Jan00] Maarten Jansen. *Wavelet Thresholding and Noise Reduction — Waveletdrempels en Ruisonderdrukking*. PhD thesis, Katholieke Universiteit Leuven, Belgium, April 2000.

- [JB99] Maarten Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for images with correlated noise. *IEEE Trans. on Image Processing*, 8(7):947–953, July 1999.
- [Jäh97] Bernd Jähne. *Digitale Bildverarbeitung*. Springer, Berlin, Heidelberg, 1997.
- [Jus86] L. Jussim. Self-fulfilling prophecies: A theoretical and integrative review. *Psychological Review*, 1986.
- [Ker98] Michael Kerres. *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*. Oldenbourg, München, Germany, 1998.
- [KK98] Christoph Kuhmünch and Gerald Kühne. Efficient Video Transport over Lossy Networks. Technical Report TR 7–1998, Dept. for Mathematics and Computer Science, Universität Mannheim, Germany, April 1998.
- [KKSH01] Christoph Kuhmünch, Gerald Kühne, Claudia Schremmer, and Thomas Haenselmann. A Video-scaling Algorithm Based on Human Perception for Spatio-temporal Stimuli. In *Proc. SPIE Multimedia Computing and Networking*, pages 13–24, San Jose, CA, January 2001.
- [Kra00] Susanne Krabbe. *Studienarbeit: Still Image Segmentation*. Universität Mannheim, Lehrstuhl Praktische Informatik IV, Dezember 2000.
- [KS00] Jelena Kovačević and Wim Sweldens. Wavelet Families of Increasing Order in Arbitrary Dimensions. *IEEE Trans. on Image Processing*, 9(3):480–496, March 2000.
- [KS01] Christoph Kuhmünch and Claudia Schremmer. Empirical Evaluation of Layered Video Coding Schemes. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 1013–1016, Thessaloniki, Greece, October 2001.
- [Kuh01] Christoph Kuhmünch. *Neue Medien für Teleteaching Szenarien*. PhD thesis, Universität Mannheim, Germany, Mai 2001.
- [KV92] Jelena Kovačević and Martin Vetterli. Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for \mathcal{R}^n . *IEEE Trans. on Information Theory, Special issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, March 1992.
- [KV95] Jelena Kovačević and Martin Vetterli. Nonseparable Two- and Three-Dimensional Wavelets. *IEEE Trans. on Signal Processing*, 43(5):1269–1273, May 1995.
- [KVK88] G. Karlsson, Martin Vetterli, and Jelena Kovačević. Nonseparable Two-Dimensional Perfect Reconstruction Filter Banks. In *Proc. SPIE Visual Communications and Image Processing*, pages 187–199, Cambridge, MA, November 1988.
- [L3] l^3 — Kooperationsprojekt Lebenslanges Lernen. <http://www.l-3.de>.
- [ICB01] Patrick le Callet and Dominique Barba. Image Quality Assessment: From Site Errors to a Global Appreciation of Quality. In *Proc. International Picture Coding Symposium*, pages 105–108, Seoul, Korea, April 2001.

- [LGOB95] M. Lang, H. Guo, J.E. Odegard, and C.S. Burrus. Nonlinear processing of a shift invariant DWT for noise reduction. *SPIE, Mathematical Imaging: Wavelet Applications for Dual Use*, April 1995.
- [Lim83] J.S. Lim, editor. *Speech Enhancement*. Signal Processing Series. Prentice–Hall, 1983.
- [LMR98] Alfred Karl Louis, Peter Maaß, and Andreas Rieder. *Wavelets*. B.G. Teubner, Stuttgart, 1998.
- [LO79] J.S. Lim and A.V. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proc. IEEE*, 67:1586–1604, December 1979.
- [MAFG82] Jean Morlet, G. Arens, I. Fourceau, and D. Giard. Wave Propagation and Sampling Theory. *Geophysics*, 47(2):203–236, 1982.
- [Mal87] Stéphane Mallat. A Compact Multiresolution Representation: The Wavelet Model. *IEEE Computer Society Workshop on Computer Vision*, 87:2–7, 1987.
- [Mal89] Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- [Mal98] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, 1998.
- [MAWO⁺97] Kurt Maly, Hussein Abdel-Wahab, Michael C. Overstreet, Christian Wild, Ajay Gupta, Alaa Youssef, Emilia Stoica, and Ehab Al-Shaer. Interactive Distance Learning and Training over Intranets. *IEEE Journal of Internet Computing*, 1(1):60–71, 1997.
- [MB95] Eric N. Mortensen and William A. Barret. Intelligent Scissors for Image Composition. In *ACM Proc. on Computer Graphics*, pages 191–198, Los Angeles, CA, August 1995.
- [McC96] Steven McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, CA, 1996.
- [MCL98] Detlev Marpe, Hans L. Cycon, and Wu Li. A Complexity–Constrained Best–Basis Wavelet Packet Algorithm for Image Compression. *IEE Proceedings on Vision, Image and Signal Processing*, 145(6):391–398, December 1998.
- [MCTM94] T. Mayes, L. Coventry, A. Thomson, and R. Mason. Learning through Telematics: A Learning Framework for Telecommunication Applications in Higher Education. Technical report, British Telecom, Martlesham Heath, 1994.
- [Mes61] Albert Messiah. *Quantum Mechanics*, volume 1. North–Holland, Amsterdam, Netherlands, 1961.
- [Mey87] Yves Meyer. Principe d’Incertitude, Bases Hilbertiennes et Algèbres d’Opérateurs. *Séminaire Bourbaki*, 145/146:209–223, 1987.
- [Mey92] Yves Meyer. *Wavelets and Operators*, volume 37. Cambridge Studies in Advanced Mathematics, Cambridge, UK, 1992.

- [Mey93] Yves Meyer. *Wavelets: Algorithms and Applications*. SIAM, Philadelphia, PA, 1993.
- [MFSW97] Michael Merz, Konrad Froitzheim, Peter Schulthess, and Heiner Wolf. Iterative Transmission of Media Streams. In *Proc. ACM International Multimedia Conference*, pages 283–290, 1997.
- [MH80] David Marr and Ellen Hildreth. Theory of Edge Detection. *Proc. Royal Society of London*, B 207:187–217, 1980.
- [MH92] Stéphane Mallat and W.L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 32(2):617–643, March 1992.
- [Mon91] S. Montresor. *Étude de la transformée en ondelettes dans le cadre de la restauration d'enregistrements anciens et de la détermination de la fréquence fondamentale de la parole*. PhD thesis, Université du Maine, Le Mans, 1991.
- [MPFL97] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, and Didier J. LeGall. *MPEG Video Compression Standard*. Chapman & Hall., New York, 1997.
- [Mul85] K.T. Mullen. The Contrast Sensitivity of Human Colour Vision to Red–Green and Blue–Yellow Chromatic Gratings. *Journal of Physiology*, 359:381–400, 1985.
- [Mur88] Romain Murenzi. *Wavelets*. Springer, Berlin, Heidelberg, New York, 1988.
- [Nas96] Guy P. Nason. Wavelet shrinkage by cross-validation. *Journal of the Royal Statistical Society, Series B*, 58:463–479, 1996.
- [Ohm95] Jens-Rainer Ohm. *Digitale Bildcodierung. Repräsentation, Kompression und Übertragung von Bildsignalen*. Springer, 1995.
- [Par96] James R. Parker. *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, 1996.
- [PFE96] Silvia Pfeiffer, Stefan Fischer, and Wolfgang Effelsberg. Automatic Audio Content Analysis. In *Proc. ACM International Multimedia Conference*, pages 21–30, Boston, MA, November 1996.
- [Pfe99] Silvia Pfeiffer. *Information Retrieval aus digitalisierten Audiospuren von Filmen*. PhD thesis, Universität Mannheim, Germany, März 1999.
- [PM93] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.
- [Poy96] Charles A. Poynton. *A Technical Introduction to Digital Video*. John Wiley & Sons, 1996.
- [RBH74] R. Rosenthal, S.S. Baratz, and C.M. Hall. Teacher behavior, teacher expectations, and gains in pupils' rated creativity. *Journal of Genetic Psychology*, 124(1):115–121, 1974.
- [Ren97] A. Renkl. *Lernen durch Lehren: Zentrale Wirkmechanismen beim kooperativen Lernen*. Deutscher Universitäts-Verlag, 1997.

- [RF85] A.R. Robertson and J.F. Fisher. Color Vision, Representation and Reproduction. In K.B. Benson, editor, *Television Engineering Handbook*, chapter 2. McGraw Hill, New York, NY, 1985.
- [RKK⁺99] Manojit Roy, V. Ravi Kumar, B.D. Kulkarni, John Sanderson, Martin Rhodes, and Michel van der Stappen. Simple denoising algorithm using wavelet transform. *AIChE Journal*, 45(11):2461–2466, 1999.
- [Roa96] Curtis Roads. *The Computer Music Tutorial*. MIT Press, 1996.
- [SCE00a] Athanassios N. Skodras, Charilaos A. Christopoulos, and Touradj Ebrahimi. JPEG2000: The Upcoming Still Image Compression Standard. In *11th Portuguese Conference on Pattern Recognition*, pages 359–366, Porto, Portugal, May 2000.
- [SCE00b] Athanassios N. Skodras, Charilaos A. Christopoulos, and Touradj Ebrahimi. JPEG2000 Still Image Coding System: An Overview. *IEEE Trans. on Consumer Electronics*, 46(4):1103–1127, November 2000.
- [Sch01a] Julia Schneider. *Studienarbeit: Multiskalenanalyse*. Universität Mannheim, Lehrstuhl Praktische Informatik IV, Dezember 2001.
- [Sch01b] Claudia Schremmer. Decomposition Strategies for Wavelet–Based Image Coding. In *Proc. IEEE International Symposium on Signal Processing and its Applications*, pages 529–532, Kuala Lumpur, Malaysia, August 2001.
- [Sch01c] Claudia Schremmer. Empirical Evaluation of Boundary Policies for Wavelet–based Image Coding. In Yuan Y. Tang, Victor Wickerhauser, Pong C. Yuen, and Chun hung Li, editors, *Wavelet Analysis and Its Applications*, number 2251 in Springer Lecture Notes in Computer Science, pages 4–15, Hong Kong, China, December 2001.
- [Sch01d] Claudia Schremmer. Wavelets — From Theory to Applications. Tutorial presented at the International Symposium on Signal Processing and its Applications, Kuala Lumpur, Malaysia, August 2001.
- [Sch02] Claudia Schremmer. Empirical Evaluation of Boundary Policies for Wavelet–based Image Coding. In *Springer Lecture Notes of Artificial Intelligence*. Springer, 2002. (accepted for publication).
- [SDS96] Eric A. Stollnitz, Tony D. Deroose, and David H. Salesin. *Wavelets for Computer Graphics. Theory and Applications*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996.
- [SE00a] Diego Santa–Cruz and Touradj Ebrahimi. A Study of JPEG2000 Still Image Coding Versus Other Standards. In *Proc. 10th European Signal Processing Conference*, volume 2, pages 673–676, Tampere, Finland, September 2000.
- [SE00b] Diego Santa–Cruz and Touradj Ebrahimi. An analytical study of JPEG2000 functionalities. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 49–52, Vancouver, Canada, September 2000.

- [SE01] Claudia Schremmer and Christoph Esser. Simulation of the Wavelet Transform on Still Images. <http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/multimedia/wavelet/WaveletDemo.html>, 2001.
- [SEK01] Claudia Schremmer, Christoph Esser, and Christoph Kuhmünch. A Wavelet Transform Applet for Interactive Learning. Technical Report TR 4–2001, Dept. for Mathematics and Computer Science, Universität Mannheim, Germany, February 2001.
- [SEL⁺99] Diego Santa-Cruz, Touradj Ebrahimi, Mathias Larsson, Joel Askelöf, and Charilaos Christopoulos. Region-of-interest Coding in JPEG2000 for interactive client/server applications. In *Proc. 3rd IEEE Workshop on Multimedia Signal Processing*, pages 389–394, Copenhagen, Denmark, September 1999.
- [SHB00] Claudia Schremmer, Thomas Haenselmann, and Florian Bömers. Wavelets in Real-Time Digital Audio Processing: A Software For Understanding Wavelets in Applied Computer Science. In *Proc. Workshop on Signal Processing Applications*, Brisbane, Australia, December 2000.
- [SHB01] Claudia Schremmer, Thomas Haenselmann, and Florian Bömers. A Wavelet-Based Audio Denoiser. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 145–148, Tokyo, Japan, August 2001.
- [SHE00] Claudia Schremmer, Volker Hilt, and Wolfgang Effelsberg. Erfahrungen mit synchronen und asynchronen Lehrszenarien an der Universität Mannheim. *Praxis der Informationsverarbeitung und Kommunikation*, 23(3):121–128, 2000.
- [SHF01] Claudia Schremmer, Holger Horz, and Stefan Fries. Testing the Knowledge Gained in Multimedia-enhanced Learning. In *Proc. Bringing Information Technologies to Education*, Eindhoven, Netherlands, November 2001.
- [SHH01] Yuta Sugimoto, Takayuki Hamamoto, and Seiichiro Hangai. Subjective and Objective Evaluation of Degraded Images Attacked by StirMark. In *Proc. International Picture Coding Symposium*, pages 121–124, Seoul, Korea, April 2001.
- [SK01] Claudia Schremmer and Christoph Kuhmünch. Simulation applets for *Multimedia Technology and Computer Networks*. <http://www.informatik.uni-mannheim.de/informatik/pi4/stud/animationen/>, 1998–2001.
- [SKE01a] Claudia Schremmer, Christoph Kuhmünch, and Wolfgang Effelsberg. Layered Wavelet Coding for Video. In *Proc. International Packet Video Workshop*, page 42ff., Kyongju, Korea, April/May 2001.
- [SKE01b] Claudia Schremmer, Christoph Kuhmünch, and Christoph Esser. Wavelet Filter Evaluation for Image Coding. Technical Report TR 6–2001, Dept. for Mathematics and Computer Science, Universität Mannheim, Germany, March 2001.
- [SKW01] Claudia Schremmer, Christoph Kuhmünch, and Holger Wons. Simulations in Interactive Distance Learning. In *Proc. 3rd International Conference on New Learning Technologies*, pages 4.1.6–4.1.7, Fribourg, Switzerland, September 2001.

- [SPB⁺98] Sylvain Sardy, Donald B. Percival, Andrew G. Bruce, Hong-Ye Gao, and Werner Stuetzle. Wavelet Shrinkage for Unequally Spaced Data. Technical report, MathSoft, Inc., Seattle, WA, April 1998.
- [Ste98] Alexander Steudel. *Das unscharfe Paradigma in der modernen Bildcodierung*. PhD thesis, Technische Universität Darmstadt, Germany, 1998.
- [Ste00] Gabriele Steidl. *Vorlesungsskriptum zur Vorlesung 'Wavelets'*. Universität Mannheim, Institut für Mathematik, 2000.
- [Str97] Tilo Strutz. *Untersuchungen zur skalierbaren Kompression von Bildsequenzen bei niedrigen Bitraten unter Verwendung der dyadischen Wavelet-Transformation*. PhD thesis, Universität Rostock, Germany, May 1997.
- [Str00] Tilo Strutz. *Bilddatenkompression*. Vieweg Praxiswissen, Braunschweig, Wiesbaden, November 2000.
- [Swe88] J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, 1988.
- [Swe94] J. Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312, 1994.
- [Swe96] Wim Sweldens. Wavelets and the lifting scheme: A 5 minute tour. *Zeitschrift für Angewandte Mathematik und Mechanik. Applied Mathematics and Mechanics*, 76 (Suppl. 2):41–44, 1996.
- [Tau00] David Taubman. High Performance Scalable Image Compression with EBCOT. *IEEE Trans. on Image Processing*, 9(7):1158–1170, July 2000.
- [TCZ96] W. Tan, E. Cheng, and Avidesh Zakhori. Real-time Software Implementation of Scalable Video Codec. In *Proc. IEEE International Conference on Image Processing*, pages 17–20, Lausanne, Switzerland, September 1996.
- [Tie99] Jens Tietjen. Hierarchische Kodierung von Videoströmen. Master's thesis, University of Mannheim, July 1999.
- [TK93] David B.H. Tay and N.G. Kingsbury. Flexible Design of Multidimensional Perfect Reconstruction FIR 2-Band Filters using Transformations of Variables. *IEEE Trans. on Image Processing*, 2(4):466–480, October 1993.
- [ULI] ULI — Kooperationsprojekt Universitärer Lehrverbund Informatik. <http://www.uli-campus.de>.
- [VBL95] John D. Villasenor, Benjamin Belzer, and Judy Liao. Wavelet Filter Evaluation for Image Compression. *IEEE Trans. on Image Processing*, 2:1053–1060, August 1995.
- [vC93] Christoph von Campenhausen. *Die Sinne des Menschen: Einführung in die Psychophysik der Wahrnehmung*. Thieme, Stuttgart, New York, 2nd edition, 1993.
- [vdB96] Christian J. van den Branden Lambrecht. *Perceptual Models and Architectures for Video Coding Applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, 1996.

- [VIR01] Cooperation Project ‘Virtuelle Hochschule Oberrhein’ VIROR. Universities Freiburg, Heidelberg, Karlsruhe, and Mannheim. <http://www.viror.de/en/>, 1998-2001.
- [vNB67] F.I. van Ness and M.A. Bouman. Spatial Modulation Transfer in the Human Eye. *Journal of the Optical Society of America*, 57(3):401–406, 1967.
- [Wan95] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates Inc, Sunderland, MA, 1995.
- [Wat95] John Watkinson. *The Art of Digital Audio*. Focal Press, Oxford, London, Boston, 2nd edition, 1995.
- [Wee98] Arthur R. Weeks. *Fundamentals of Electronic Image Processing*. SPIE/IEEE Series on Imaging Science & Engineering, 1998.
- [Wei99] Joachim Weickert. *Vorlesungsskriptum zur Vorlesung ‘Anwendungen partieller Differenzialgleichungen in der Bildverarbeitung’*. Universität Mannheim, 1999.
- [Wic98] Mladen Victor Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A.K. Peters Ltd., Natick, MA, 1998.
- [Wie49] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*. MIT Press, 1949.
- [Win00] Stefan Winkler. *Vision Models and Quality Metrics for Image Processing Applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, December 2000.
- [WJP⁺93] Arthur A. Webster, Coleen T. Jones, Margaret H. Pinson, Stephen D. Vorna, and Stephen Wolf. An objective video quality assessment system based on human perception. In *SPIE Human Vision, Visual Processing, and Digital Display IV*, volume 1913, pages 15–26, San Jose, CA, February 1993.
- [WM01] Mathias Wien and Claudia Meyer. Adaptive Block Transform for Hybrid Video Coding. In *Proc. SPIE Visual Communications and Image Processing*, pages 153–162, San Jose, CA, January 2001.
- [Won00] Holger Wons. *Studienarbeit: Kosinus und Fourier Transformation*. Universität Mannheim, Lehrstuhl Praktische Informatik IV, Mai 2000.
- [ZGHG99] Philip George Zimbardo, Richard J. Gerrig, and Siegfried Hoppe-Graff. *Psychologie*. Springer, Berlin, Heidelberg, 1999.